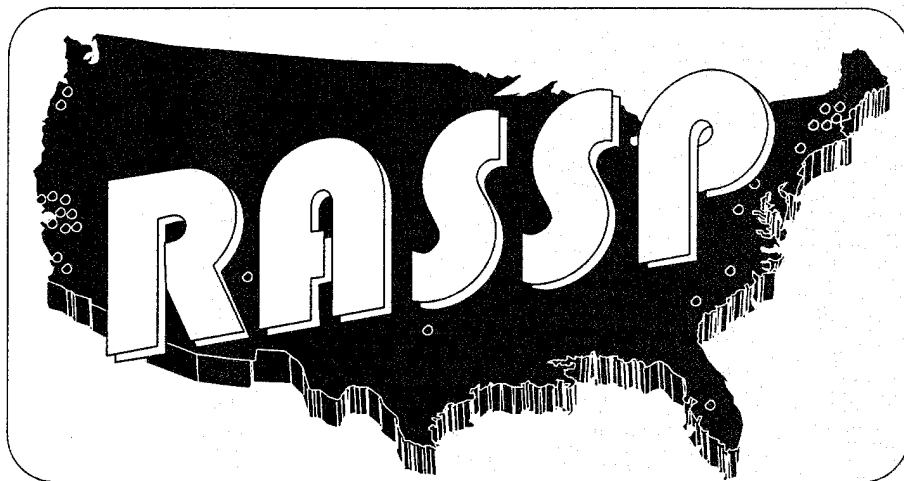DTIC
SELECTE
JAN 2 5 1995
C D

(0)

Rapid Prototyping of Application Specific Signal Processors (RASSP)

**RDE
System
Description
Final
(3rd)**

22 December 1995

Advanced Research Projects Agency
Naval Research Laboratory

19960122 022

SANDERS
A Lockheed Martin Company

Document #AVY-L-R-00252-101-A

DTIC QUALITY INSPECTED 1

Rapid Prototyping of Application Specific Signal Processors (RASSP)

**RDE
System
Description
Final
(3rd)**

22 December1995

Submitted by:	Lockheed Sanders, Inc. Hughes Aircraft Motorola ISX Corporation
Under Contract:	N00014-93-C-2172
As:	CLIN 0004AG, Seq. No. A007
For:	Advanced Research Projects Agency 3701 North Fairfax Drive Arlington, VA 22203-1714
	Naval Research Laboratory 4555 Overlook Avenue Washington, DC 20357-5347

SANDERS
A Lockheed Martin Company

P.O. Box 868
Nashua, NH 03061-0868

Document #AVY-L-R-00252-101-A

Table of Contents

List of Figures

List of Tables

1   Purpose

The purpose of this document is to describe the Lockheed Sanders Team RASSP Design Environment (RDE). The RDE provides a working environment to support users of the RASSP process to develop signal processing systems more quickly and with greater quality than was previously achievable. The RDE is a CAD tool-independent process management system that supports distributed development. This document describes the RDE and the components that make up the environment.

The predecessor of this document was the Annual CAD System Description Document. It was renamed this year to be the RDE System Description Document because the focus of the Sanders' RASSP program is on the RDE and the building of an environment that supports distributed development. The CAD tools used by the Demonstration and Benchmark teams are commercial tools that are widely known by the electronic design developers, so are not described in this document. Instead, this document completely describes the development and features of the RDE.

2   RASSP Design Environment

The RDE is a set of commercial and RASSP developed components that facilitate the development of electronic systems in a distributed environment. The RDE supports a heterogeneous computing environment, links between geographically diverse locations, tailorable configuration management, and exchange of product information between the many varied disciplines. This supports an improved product development process allowing for rapid iterations, incremental promotion, and scalable configuration management controls.

This section will provide the physical description of the RDE and discuss the RDE software components. The RDE is being developed iteratively with a series of builds. The RDE is scheduled for 10 builds during the four years of the RASSP program. The development of each build is three to five months in duration. At the end of each build, the components are integrated with each other and released to each of the team members to use. The requirements for the next build are developed based on the long term vision of the RDE and from feedback from the users of the RDE. This document represents information from Build 5 of the RDE.

2.1   Physical RDE Architecture

The RDE was designed to support concurrent development of signal processing systems. This environment allows a team of engineers and managers to work together on a project from geographically different sites and using different platforms. The RDE hardware architecture consists of heterogeneous computers and high-speed data communication networks. It is a geographically distributed architecture consisting of Unix workstations, file servers, IBM compatible PCs, and McIntoshes, all interconnected via communication networks. Groups within companies are electronically connected by a Local Area Network (LAN), and are connected to other groups or companies over a Wide Area Network (WAN) using Internet. The largest issue about connected WANs together is to overcome the security issues to protect one's data from unauthorized access. The details of how the Sanders RASSP network is set up is described in Section 6.

Currently, we have been focusing on Sun Unix stations as the preferred platform using the X-Windows system. The current RDE is running on SunOS 4.x, but is being ported to Solaris. There are long term plans to port to HP. As the RDE is ported to each platform, the previous platforms will also be supported. When the RDE has been ported to Solaris, it will be made COSE Common Desktop Environment (CDE) compliant.

Future plans are being made to provide many of the RDE services using a World Wide Web (WWW) browser such as Netscape. This will immediately allow us to achieve the goal of having a multi-platform environment.
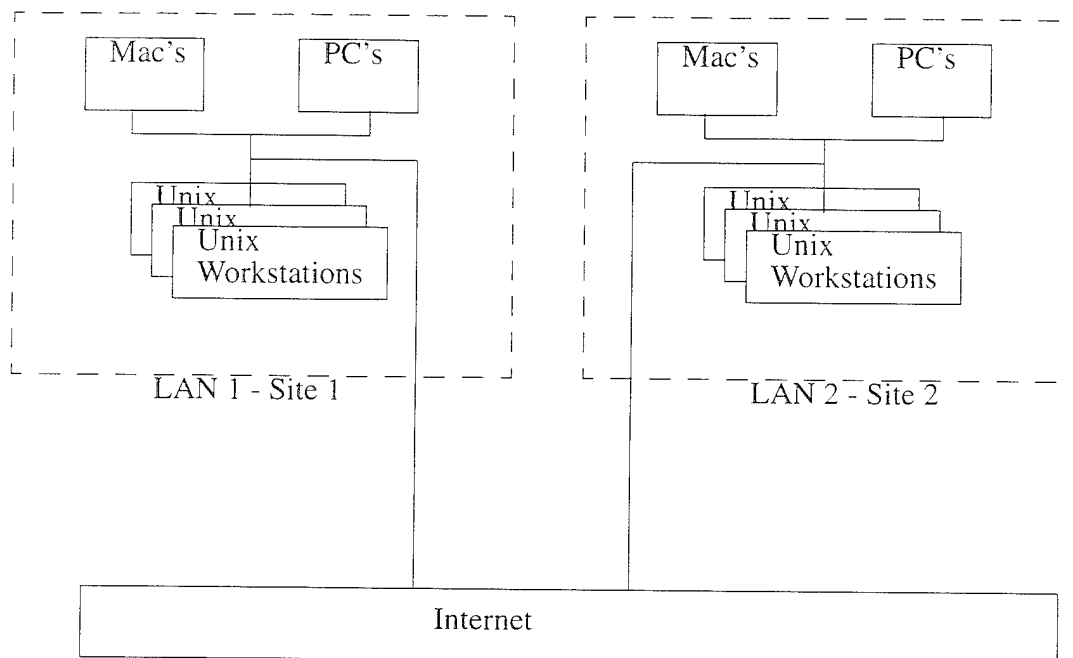
Figure 1: Physical RASSP Environment

2.2   Enterprise Framework

The Lockheed Sanders RASSP team has invested much time searching for an Enterprise Framework (EFW) that could be used as the electronic framework to the RDE. This started with an original setup of requirements that were generated by the RASSP Systems Engineering team. A Request For Information was sent to vendors who were thought to have a solution to the EFW. From the nine vendors that responded, four were selected for a product demonstration and discussion with them. Two vendors were downselected for an in-house evaluation of their product. The result was that the best met only 50% of the original requirements based on the test procedure that

was developed. Since that time, we have been putting together the components from commercial utilities as well as developing our own services where commercial solutions were not available.

The EFW focused on five areas as shown in Figure 2. These areas are
- Data Management
- Workflow Management
- User Interface
- Communications Management
- Tool Encapsulation
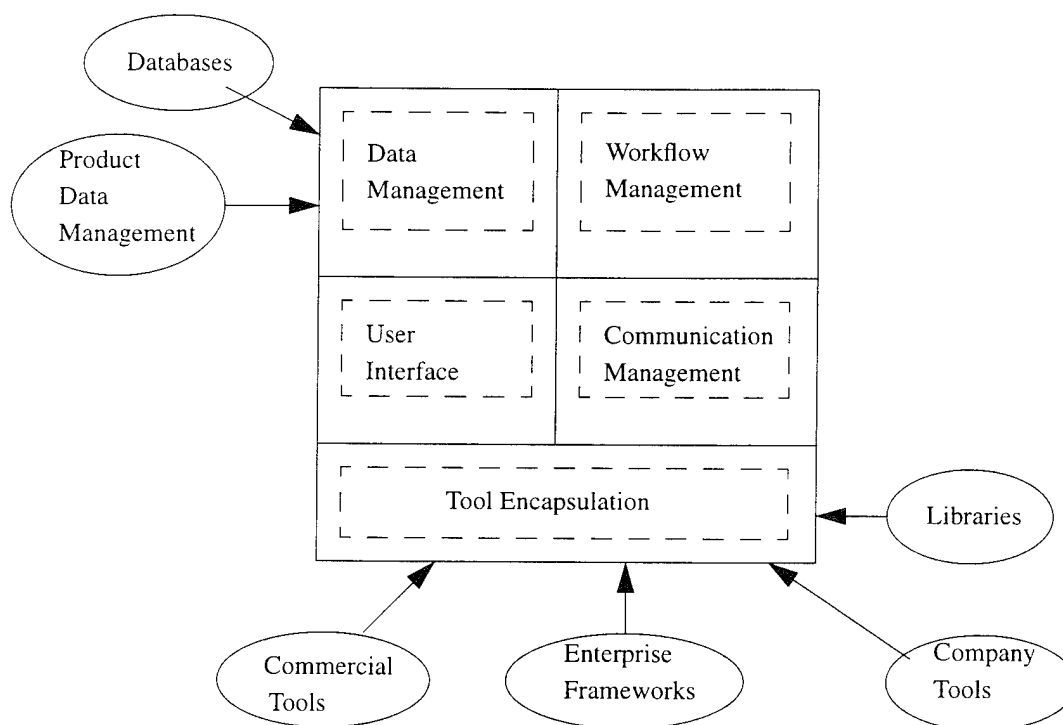
Figure 2: Original RASSP Enterprise Framework

The RDE has most recently focused in two areas, Process Management and Data Management. Process Management has grown out of Workflow Management as our idea of process has evolved. Process Management contains several parts of which Workflow Management is one. RASSP is a process program and a section of this document is devoted to Process Management.

The other major area that we are focusing on is Data Management. A complete description of Data Management is described in this document. This section describes the current RDE.

3 Current RASSP Design Environment - Build 5

This is the fifth of ten scheduled builds of the RDE. It is the goal of the RDE to support the development of signal process systems with a geographically distributed team. All the components that make up the RDE have this in mind, and support distributed control or access.

The RDE is comprised of a series of components and services. The RDE has a major focus on Process Management and Data Management. The following sections will briefly describe the major components of the RDE. A more complete description of these components can be found in the "RDE User's Guide" of Appendix B.

Table 1:

General Services	Process Management	Data Management
Component Communications	Task Manager	Product Structure Editor
Desktop	Metric Browser	Remote Data Access
Problem Reporting	Review Utility	Source Reuse Utility
User Support Utility		

Communication utilities and services that are public domain or can be obtained commercially are discussed in the Network Communications section of this document, Section 6.

3.1 Component Communications

The communications between the RDE components is done by sending ToolTalk messages to an RDE bulletin board. Any component in a session can post a message to the bulletin board. If another component has subscribed to that message, it will be notified with a ToolTalk message. As a new component enters the session, it can query the bulletin board for the desired information.

The RDE will use COSE ToolTalk as the messaging system to communicate between components of the RDE. Some messages are RDE defined, and others are standard messages defined by CFI. The ability to send messages to commercial tools depends on whether the tool supports the standard messages. The messages that the RDE currently uses are listed below.

- Current project
- User
- Current working directory
- Current process phase

Other messages that will be used when converting to COSE CDE are listed below.

- Launch Tool
- Tool successfully initialized
- Tool has finished execution
- Check file into database

3.2   Desktop

The RDE Desktop Utility provides a desktop strip menubar which allows users to navigate the RDE, access RDE Utilities and Services, and launch encapsulated COTS Tools.

The RDE is an environment shell in which working conditions can be customized and tools can be accessed and launched. The way that categories are set up is remembered from session to session, in regards to the categories that are defined, which tools are in those categories, and the position and size of those category tool palettes.

3.3   Problem Reporting

The Problem Report Utility provides the ability to submit problem reports for use by the Problem Management System. A key challenge of development teams is to effectively identify, contain, and resolve development problems. The Problem Management System facilitates the handling of problems throughout the product development lifecycle. Remote access to the Problem Management System facilitates problem management for the geographically diverse IPPDT and the Virtual Corporation. Problem Report data is available for metrics collection and display. Reporting is quick and easy and the system automatically informs the responsible parties and sends appropriate problem notifications.

3.4   User Support Utility

The User Support Utility is a WWW based page that contains information for the RDE user. This information is a set of hyperlinks to information including bug alerts, RDE Users Guide, and newsgroup interface for group RDE discussions. This web page can be accessed from the RDE help menu.

3.5   Task Manager

The Task Manager is a tool that is used by engineers and managers on a project to keep track of and report on their tasks. The task manager can assign the activities of a schedule to individuals on a project, and allow tasks and subtasks to be created. Each user has their own list of tasks, and is notified when new tasks are assigned to them. When a task is completed, the use indicates that the activity is done. A report of the current status of the project can be viewed at any time by managers or engineers.

3.6   Metric Browser

The Multi-Metric Browser Utility provides the ability to access and display metric data. The RDE automates data collection and metric generation. Customizing metrics provide any IPPDT member with useful data and feedback. The Multi-Metric Browser retrieves data on request from the RDE database to populate the desired users form or report.

The database is populated by the metric generators. Metric generators are normally run as a cron job. Currently, four major metric reports are available through the browser: code size, number of open problem reports, number of closed problem reports, and review packets created.

3.7   Review Tool

The Review Utility supports and automates the entire review process. Consistently accurate Technical Peer Reviews help ensure compliance with the development process and a mature, quality product. Review data is encapsulated in a Review Package, maintained throughout the review process. The Review Utility facilitates on-line review by geographically dispersed reviewers. The review leader and all the reviewers are selected through the utility. The Red Lining feature displays the markups from all the reviewers and allows for new markups. By having different colors represent each reviewer, it is easy to see and identify and to block out any of the comments or markups made. The review members are notified through e-mail of the status of their review packages. Utility data is also available for metrics collection and display.

3.8   Product Structure Editor

The Product Structure Editor provides the ability to structurally organize the elements of a product. It allows the graphical representation of the hierarchy of a product including source code, documentation, and images and stores the data and the mapping of the data into a database. The users define all the object types, actions on those types, tools for those actions, formats for the tools, and possible relationships between objects. The structures can be created in the hierarchy view, which displays the structure in a parent-self-children format. Other views are available to represent the structure, they include an information view, which displays the information of the center element, and a relation view, which displays the relationship of the elements in the structure.

This utility provides an organized method of maintaining the product data over its lifecycle. Source and document versions are handled through the check in and check out options. The database itself can be backed up through the export/import options. And, the database information on the structure can be easily viewed by all interested individual at all stages of development.

3.9   Remote Data Access

The Remote Data Access Utility(RDA) provides access to remote configuration-controlled data repositories which house the distributed RDE database. This multi-platform client/server software for accessing the RASSP database currently runs on Macintosh, Unix, and Windows platforms.

The RDA consists of a Server, a Service Broker, and a Client. The Server registers itself with the broker and waits for a client. The broker maintains the list of available servers. When a client asks the broker for a server and receives the necessary information to connect to one, the server then disconnects itself from the broker and spawns a new server. The new server registers itself with the broker and waits for the next client.

3.10 Source Reuse Utility

The Reuse Utility provides the ability to populate, query, and update the reuse database. The reuse of components from previous development efforts helps reduce cycle time. Reuse Management capabilities and remote database synchronization facilitate reuse for the IPPDT and the Virtual Corporation.

The rest of the original five areas of the EFW, User Interface, Communications Management, and Tool Encapsulation, are general services that are common to both the Data Management and Process Management areas. The only work continuing in the general services area will be to support the needs of Process Management and Data Management.

4 Process Management

This service facilitates the encapsulation and integration of tools for inclusion in the RDE. Tools will be encapsulated without requiring the vendor to change their tool. This enables a heterogeneous set of tools to work together.

4.1 Process Management Vision

Process Management provides automated, distributed capabilities for presenting the RASSP process on-line, assisting users in executing the process, and tracking progress against the process. It also provides integrated capabilities for collecting, analyzing, and displaying process and product metrics. Product developers and managers will have capabilities for event notification, task statusing, and process compliance assistance and assessment. They will also have capabilities for dynamically viewing history and actuals against plan and requirements. These automated, distributed capabilities will increase situation awareness and improve decision making by enhancing the quality and timeliness of communication. By providing a minimum level of discipline, they will also facilitate concurrent development. This will reduce wait time, avoid unnecessary effort, and result in reduced cycle time. Furthermore, the capabilities for tracking and measuring progress can provide process improvement insights.

4.2 Process Management Paradigms

There are two basic process management paradigms, routing and enactment. RASSP focuses on the enactment paradigm. However, since routing is essentially a constrained subset of enactment, RASSP actually focuses on both.

4.2.1 Routing

The Routing Paradigm is used primarily in financial, business, and administrative environments. Routing is applied to paper and electronic "forms" or "documents". Some examples are problem reports, purchase orders, change requests, and technical reports. Typically, each form is routed sequentially through a well-defined series of steps that include conditionals to handle different cases. Not accommodated, however, are process innovation, process iteration, and the addition of new steps in response to changing conditions.

Routing is characterized by a fixed set of steps, each of which is specific and well defined (what is performed and how it is performed). Also, all "forms" of a given type follow the same process. Although there may be many possible routes for a form to follow through the process, the process itself is static. It is somewhat analogous to a map since one can follow many routes but cannot change the existing roads or cities. This paradigm is appropriately rigid about how "forms" are processed.

4.2.2   Enactment

The Enactment Paradigm is employed in engineering and other creative environments. Enactment is applied to product designs and other conceptual objects. Some examples are electronic products, software tools, and mechanical devices. Typically, each object is developed by following a tailored set of activities (a process) formulated to meet the object's functional and performance requirements. An object evolves as it moves through the process, starting in a highly abstract form and ending in a fully realized form. Furthermore, objects move through the activities in a non-deterministic way, making many local iterations based on innovative choices and experimental results. Finally, decisions about which activities are present within the process and the details of what is performed during an activity are often made at the time a sub-process or activity is initiated, since only then is enough information known to determine such details. Thus, for enactment, a process can be dynamic.

Enactment is characterized by a set of activities, each of which is defined by sets of entry and exit criteria, and an added-value transformation that is applied to the inputs to create the outputs. Typically, the transformation includes synthesis and validation steps. The specific set of steps and their order of application for accomplishing an activity is not necessarily specified. Essentially, each activity is a well defined, well-bounded problem with a clear definition of "done", with the exact method of solving the problem left to the performer. This paradigm is appropriately flexible. It accommodates innovation, creativity, and experimentation (e.g., trade studies).

Enactment is also characterized by its recognition that the entire product development process cannot be known ahead of time. The process and product evolve together. Decisions made while enacting early activities determine many of the downstream activities. For example, packaging decisions that are made early in a process determine whether downstream sub-processes will be needed for module, ASIC, or FPGA design.

4.3   Process Terminology

We know of no standard terminology for describing and discussing processes. Without definitions, it is expected that the meanings of these terms would be different between individuals and would be influenced by the roles of the individuals enacting a process. Therefore, to avoid confusion and provide clarity, the words "Project", "Process", "Activity", and "Task", as used by RASSP, are defined here. They are used consistently within our documentation, presentations, and implementations (such as our web process pages). These definitions relate to experiences on RASSP and we will use RASSP examples for illustrations.

4.3.1 Project

A project is an undertaking, requiring concerted effort, that consumes resources and delivers a product. A project is more than just a process. A project contains tailored process instantiations, schedules, resources (people, tools, facilities, etc.), costs, contractual agreements and more. A given project may have several processes or sub-processes (e.g., a unit process, a board process, and a software development process).

4.3.2 Process

In simple terms, a process is a series of operations performed in the making or treatment of a product. A process can be hierarchically decomposed into smaller sub-processes down to the bottom or leaf level operations that are called activities. Sub-processes are collections of activities and other sub-processes, grouped together to simplify understanding of the whole process.

Hierarchic grouping can be performed many different ways, but as long as the set of activities does not change, all the possible hierarchic groupings are, in fact, the same process. For an example sub-process, refer to Figure 3 which contains a RASSP Benchmark 3 process diagram. Here, sub-processes (further decomposed in other diagrams) are indicated by shadowed boxes, activities are portrayed by plain boxes, and data flows are represented by arrows. Diagrams like this one do not contain a complete description of a sub-process. However, it is a convenient method for assisting understanding.
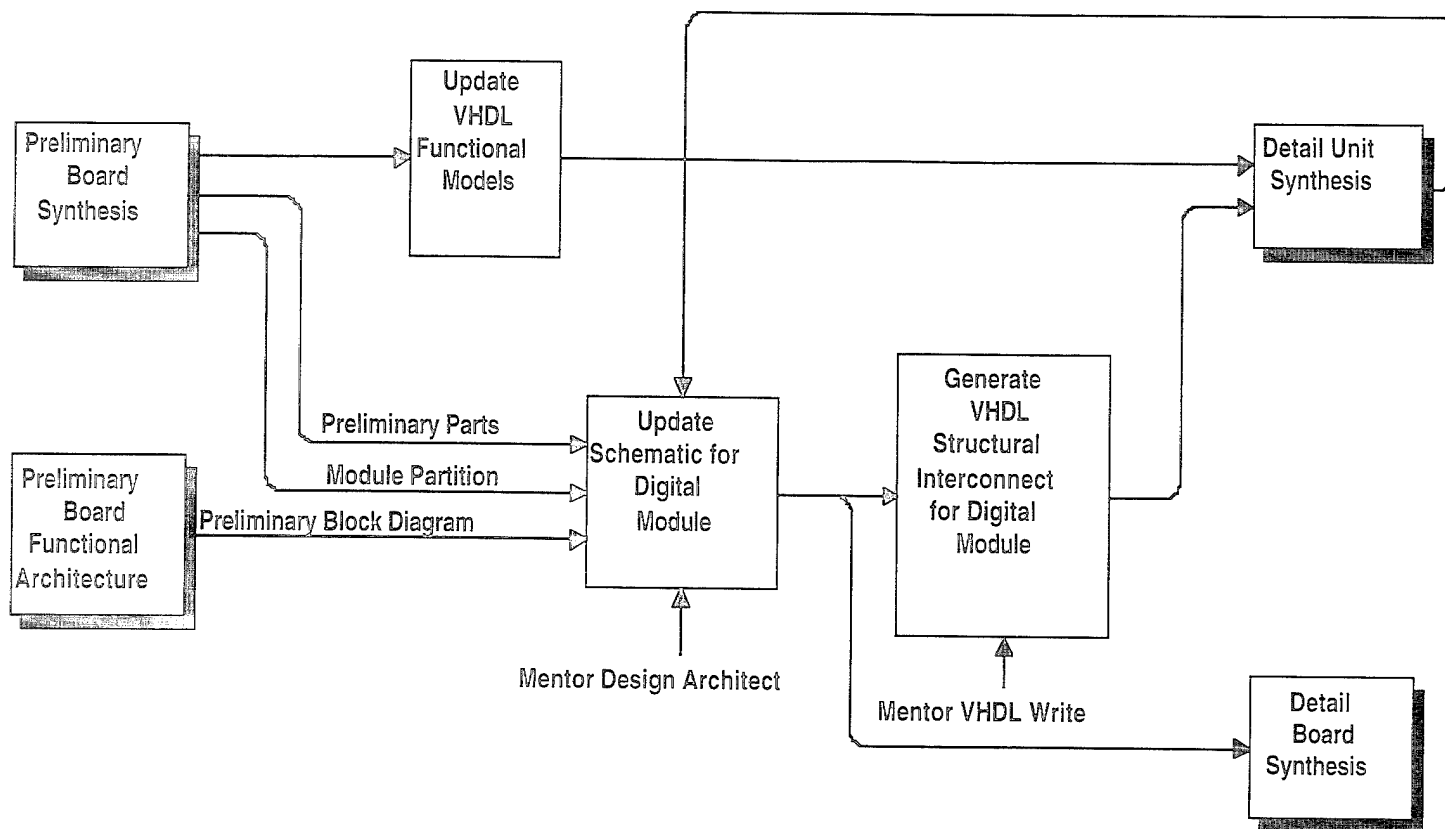
Figure 3: Example Sub-Process Diagram from RASSP Benchmark 3 Process

4.3.3 Activity

An activity is the effort represented by a leaf level operation in the hierarchic decomposition of a process. Consistent with the enactment paradigm, an activity is a well defined, well-bounded problem with a clear definition of "done". The exact method of solving the problem is left to the performer who is considered a knowledgeable, competent professional capable of performing the work without detailed, step-by-step instructions.

When formulating a process, it is sometimes difficult to know when to stop decomposing the process and settle upon the final set of leaf level activities. We refer to this problem as the "level of detail problem". Guidance for solving this problem is beyond the scope of this document. However, we have chosen an activity to be a unit of managed work, or, in other words, a well-defined unit of work that a manager wishes to track without further decomposition. Furthermore, for us, there is conceptually a one-to-one mapping of activities in the process to line-items in a project-schedule's Gantt chart. Thus, activities are the lowest level of cost accounting, metrics reporting, and data package generation of concern to the manager. Also, activities can be formulated such that their outputs are at critical boundaries where data packages are passed between organizations or software tools. These boundaries can also be phase containment boundaries, beyond which errors are considered defects.

Each activity is defined by its inputs, outputs, entry criteria, exit criteria, tool set, tasks, and performer types (e.g., roles, expertise). An activity cannot begin until its entry criteria are satisfied. An activity cannot completed until its exit criteria are satisfied.

Often, an activity's entry criteria is simply the availability of its inputs. It can also be a complex set of conditions including partial sets of inputs and preliminary versions of inputs. Often, an activity's exit criteria are the completion of all outputs and verification that quality standards have been met. Since the outputs of activities are the inputs to other activities, there can be a complex interrelationship between activities, many of which can be performed in parallel. During performance of an activity, tasks are carried out. The exact set of tasks, their order of execution, and the number of iterations through them is determined by the performer (with guidance from documentation, checklists, and lessons learned).

The state of most activities' entry criteria are determined by the state of other activities' exit criteria. Exceptions exist at the boundary or interface of the process. Examples are: arrival of parts in receiving, receipt of authorizations to increase spending, customer changes to requirements. When formulating a process, it is critically important to understand the process's interface with its operational environment.

4.3.4   Task

A task is an individual work operation, many of which must be performed to complete an activity. At the discretion and convenience of performers, tasks can be decomposed into sub-tasks. The need for some tasks can be predetermined but many cannot. Although sets of related tasks could be represented as flow diagrams, the tasks' order of performance is often unimportant and, since the need for them is determined dynamically, maintaining flows is economically unjustified. (Especially successful activities can be recorded for reference in process documents and lessons learned.)

4.4   Process Management Lifecycle

Our Process Management Life Cycle consists of continuous process usage and improvement loops. See Figure 4. Initially, reference processes for new product classes are defined. From then on, processes are iteratively improved by usage (enactment) and evaluation. For both loops, reference processes are interconnected, tailored, and expanded into a plan. The plan is either enacted by a product development effort, or evaluated through simulation. During enactment and evaluation, metrics and simulation results are collected. The metrics and simulation results are used to adjust the plan or to improve the reference processes.
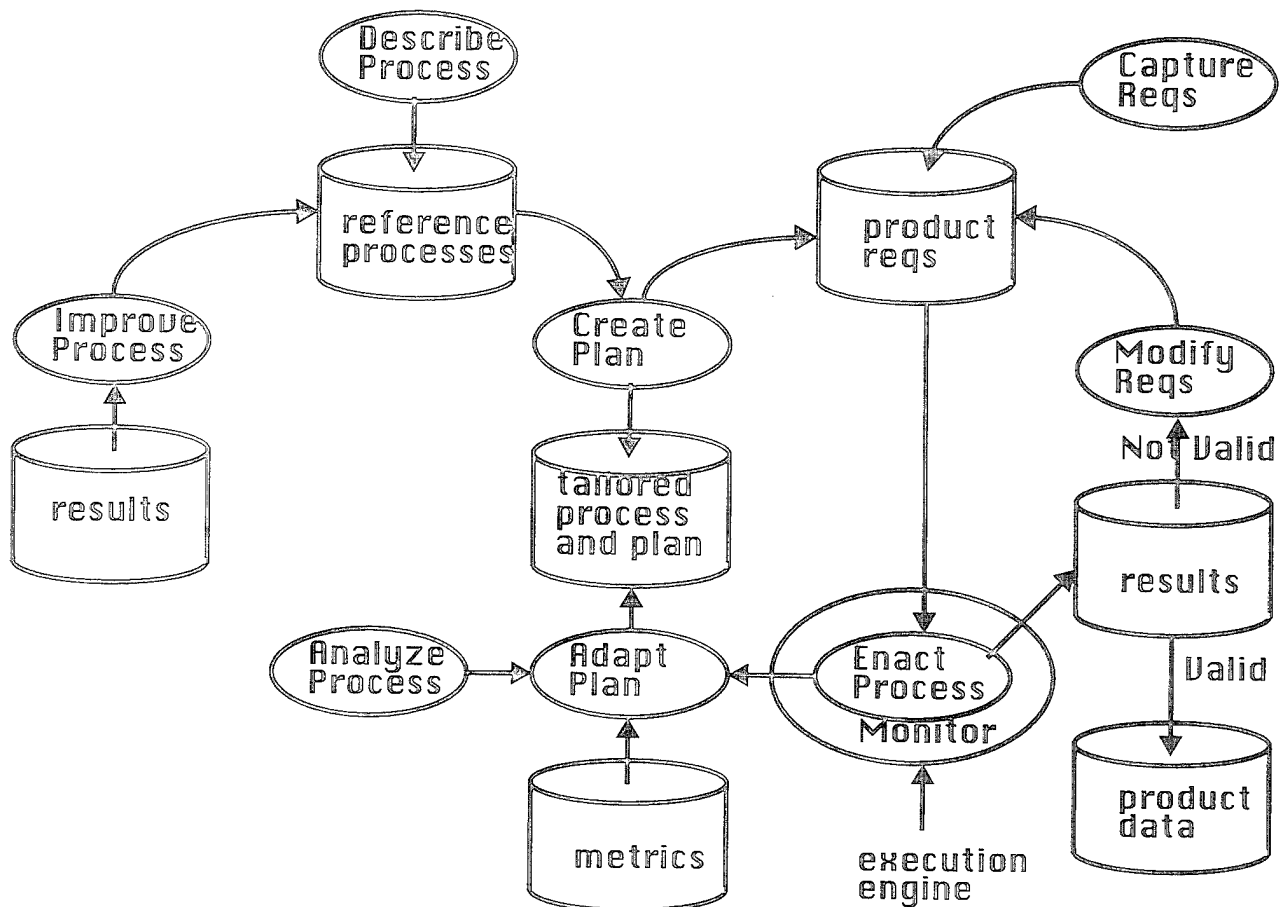
Figure 4 - RASSP Process Management Life Cycle

From our process management perspective, product development begins with planning (e.g., process composition, tailoring, scheduling, resource loading, costing) and ends with a completed product (and product data) within cost and schedule. In this perspective, we assume the existence of 1) product requirements, and 2) "reference processes" for developing products that can satisfy the requirements.

4.4.1   Describe Processes

Describing a process (or sub-process) consists of either capturing an existing process or defining a new one. Capturing an existing process consists of collecting information about a process and creating a process description. There are numerous methods for capturing process descriptions. However, such methods are outside the scope of this document. Nevertheless, the result of describing a process is the creation of a "reference process" that is placed in a library for subsequent use.

4.4.2   Reference Processes

Reference processes are processes and sub-processes that are stored in a library for use in composing product specific "tailored" processes. Reference processes are generic "if-ever" processes.

This means that reference processes address families or classes of products and that they are not specific to any particular product instance within a class. Furthermore, reference processes address all possible, known development situations that can occur for a product class. If it's ever the case that a specific product development must include activities to deal with some aspect of the product, then the reference process contains the needed activities. It is rare that any specific product development effort will need to perform all activities contained in a reference process.

Since some processes are tool and technology dependent, reference processes may contain prescribed tools and data file formats. Such prescriptions are also convenient for individuals who prefer not to specify such details and use what has proven successful in the past. Furthermore, processes may also contain product structure segments with which to organize the product data produced by process enactment. Finally, processes may contain the roles and skills necessary to carry out each activity.

The RASSP Program is developing a reference process for the development if Digital Signal Processors, DSPs. The RASSP Design Environment, RDE, will contain capabilities for tailoring this reference model for the development of specific DSPs.

4.4.3   Product Requirements

Product requirements describe the detailed specifics of the functions and performance that a particular product must satisfy. Product requirements are provided by the customer and are assumed to be somewhat incomplete at the beginning of product development. During product development, the requirements are refined by involving the customer in the product development effort. Consequently, the tailored process and plan evolve over time as the requirements mature.

4.4.3.1   Create Plan

Creating a product development plan consists of tailoring a composition of reference process, formulating a schedule (adding time and resources to the tailored process), applying appropriate constraints (e.g., availability of things, desire for events to occur at specific times), and assessing risks. Included in the creation of the plan is a balancing of product features, activities, resources, and risk so that the product satisfies the requirements and is developed within given constraints. Planning is done prior to process enactment.

A plan will not be perfect from the start. It will contain unspecified items since not enough is known at the start. Thus, the plan will need to change in real time to accommodate events.

4.4.3.2   Tailor Process

Process tailoring consists of creating a composition of several reference processes, appropriately modifying the interfaces between them, defining or adjusting the product structure with which to organize and manage generated product data, and defining actions to be taken automatically when certain events occur.

To aid tailoring, reference processes are stored in a library. They are reusable sub-processes that can be instantiated before and during process enactment. These sub-processes are templates that include candidate activities, data objects (inputs and outputs), entry and exit criteria, tools, tasks, narrative descriptions, metrics, events, and actions. Events, if supported by the design environment, cause automatic execution of actions when their activation rules are satisfied, such as notifications to be sent, software tools to execute, and status information to be collected.

4.4.3.3 Process Composition

In its simplest form, process composition consists of examining the product requirements and selectively editing a reference process to include only those activities that are appropriate for the specific product.

In its typical form, process composition consists of selecting several reference processes, editing them, interconnecting them, and ensuring that any holes that remain can be filled by other existing reference process. Holes will exist when down-stream decisions are needed before final selection can be made.

In its most complex form, process composition deals with virtual corporation formation. Significant process negotiation and editing is required when several companies are teaming, since each has its own reference processes, and it is likely that very little process consistency exists across them. Nevertheless, without a tailored process, it is unlikely that the operation of a virtual corporation will be smooth.

4.4.3.4 Tool selection

Tool selection consists of examining the software tools employed by all the activities and ensuring that 1) they are appropriately consistent across the interconnected reference processes and 2) that they are available to the project. If this is not the case, then modifications are made to the tailored process to correct any problems. This may include the inclusion of translators, the replacement of sub-processes, or re-thinking of the product's architecture.

4.4.3.5 Interface Modification

Interface modification consists of examining the internal interfaces (between sub-process) and the external interfaces (outside the process) to ensure that data objects and types are consistent and match up. If this is not the case, then appropriate adjustments are made so that information can flow freely through the tailored process and be smoothly exchanged with its process environment.

4.4.3.6 Product Structure Definition

Product structure definition consists of examining the product structures associated with the sub-processes to ensure that they are properly constructed to contain the data that the project wishes to manage. If this is not the case, then appropriate modifications are made to the product structures.

The organization of the product structures is influenced by the tools employed and the other sub-processes in the tailored process.

4.4.3.7   Event-Action Definition

Event-Action definition consists of examining the roles, notifications, and automatic actions present in the reference processes and determining if they are appropriate for the project being planned.

4.4.3.8   Create Schedule

- Time
- Cost
- Resources
- Risk

4.4.4   Enact Process

Enacting the process consists of executing the tailored process according to the project schedule. Process enactment results in the development of a product that satisfies the product requirements. During process enactment, as the product evolves -- design decision and technology choices are made -- the process is adapted to reflect those decisions. For instance, if a decision is made to use FPGA's instead of an ASIC, then necessary process changes are made, typically adding an instance of a sub-process to the plan. During enactment, which consists primarily of synthesis and verification activities, candidate designs are compared with requirements to ensure compliance.

4.4.5   Product and Data

Product and data are the result of the product development effort (enactment of the tailored process). The product is the primary deliverable. The data consists of all the managed representations of the product, organized by the product structure. Essentially, the Product data is a description of the product in sufficient detail that the product can be reproduced.

4.4.6   Monitor

Monitoring the enactment consists of collecting process and product metrics, and analyzing them to determine progress towards cost and performance objectives and to assess risk. Low level metrics are used by performers to coordinate activities. Low level metrics are also rolled up to provide management with appropriate visibility of project status and to alert them to potential problems.

4.4.7   Metrics

Metrics are measurements collected during process enactment. They include both product and process metrics. Process metrics are analyzed to determine progress against plan. Product metrics are analyzed to determine compliance with requirements. Together they are analyzed to determine

productivity. The results of these analyses are used to adapt the plan. Included in these analysis are lessons learned that are also used to improve the reference processes.

4.4.8   Adapt Plan

Adapting the plan consists of changing or updating the tailored process and project schedule as events unfold. Better methods are incorporated for accomplishing development objectives, new tools are adopted, schedules are modified, and so on. Much of the adaptation is driven by design decisions and the conclusions arrived at through metrics analysis. Simulation can be used to aid management decision making.

4.4.9   Predict Outcome

Predicting the outcome of a project consists of devising candidate changes to the tailored process, projecting each of them into the future, and selecting the best one. Predicting the outcome of a project occurs in two different situations, one during enactment, the other during process improvement.

During enactment, predicting the outcome of a project is a management activity. It will typically occur when significant events have occurred which require a change to the project plan. Here, given the current state and circumstances of the project, prediction is aimed at selecting those changes to the tailored process that achieve the best possible outcome. One possible approach for doing this consists of converting the tailored process and plan into an executable model, loading the model into a simulator, using the collected metrics as initial conditions, and running the model to conclusion. By doing this repetitively with various candidate changes to the tailored process and plan (process, resources, etc.), simulation can help predict the outcome of the various choices. Here, the role of process simulation is to provide management with powerful "what-if" capabilities that can help predict the consequences of decisions in very complex situations. For realistic results, real world actuals must be available for simulation -- e.g., production rates and windows of availability for production lines to determine what delivery commitments can be met.

During process improvement, predicting the outcome of a project is a process-developer activity. It will occur whenever the enterprise chooses to improve its reference processes. Here, prediction is aimed at determining the value (contribution to productivity and quality) of proposed changes. Again, one possible approach for doing this consists of simulating reference processes. Here, the plan (tailored process, etc.) is simulated under various anticipated conditions using representative example products (or past projects) as evaluation vehicles. The results of process simulation show the effects of various changes to tailored processes, with different product configurations, using different resource loadings, operating under varying constraints, and with different event probabilities.

4.4.10   Improve Process

Improving the process consists of 1) taking lessons learned from enactments and incorporating them into reference processes, and 2) taking tailored processes and conducting experiments to determine if proposed modifications to reference processes will result in expected results.

4.5   Process Management Requirements

The requirements enumerated here reflect our goal of providing a "Tool-Independent Process Management System" that enables Distributed Development. This means that teams composed of individuals from several companies, geographically distributed over large distances, and in different time zones, can effectively work collaboratively as a virtual corporation. In this context, process management is not constrained to include only engineering processes. It must also provide capabilities that can support administrative, financial, material (i.e., supply management), and other management (above the shop floor) processes.

4.5.1   Distributed Operation

The process management capability shall provide capabilities for operating over a combination of LAN (local area network) and WAN (wide area network) configurations. It must operate over the network using web technology since many of the process management functions will be provided through web browsers.

4.5.2   Multi-Platform

The process management capability shall be capable of operating on multiple compute platforms including Sun, HP, Mac, and PC Windows. This is because in many companies significant product development work is done on personal computers. Furthermore, most managers have Mac's or PC's, and they are the ones who have most need for monitoring the status of on-going work.

4.5.3   Maintain Process State

The process management capability shall provide capabilities for persistent maintenance of the state of each activity. Furthermore, it shall be capable of recalculating the state at any time.

The process management capability shall provide capabilities for manual setting of state values (with appropriate access control). This feature is needed when it is necessary to override the results of entry and exit criteria evaluation. In some cases, this will provide the means for process deviating.

The process management capability shall also provide capabilities for querying the state of activities so that users can ascertain progress through the process and determine what criteria must be satisfied for additional activities to complete.

4.5.4   Signal Events

The process management capability shall provide capabilities for users to signal that certain events have occurred, thus triggering automated actions to occur such as notification, state change, and metric collection.

4.5.5   Process Deviations

The process management capability shall provide capabilities for allowing users to deviate from the tailored process. This is needed because there is no assurance that a process will be perfect and include all activities required. However, all deviations will be recorded and users asked for the reasons for deviating.

4.5.6   Provide Audit Trail

The process management capability shall provide capabilities for logging all events that occur during process enactment, including adaptations of the process.

4.5.7   Monitor Process

4.5.7.1   Capture Metrics

The process management capability shall provide capabilities for capturing process and product metrics during enactment of tailored processes. The captured metrics must be traceable to the processes' activities and tasks, even when the process has been adapted to account for evolving events.

4.5.7.2   Provide Thermometers

The process management capability shall provide capabilities for displaying dynamically updating digital or graphic thermometers. The thermometers shall display the values of user selected metrics in comparison to goals, objectives, or constraints. Users shall also have the ability to display thermometers for both product and process metrics and shall have the ability to define new metrics as functions of existing ones. For instance, if one metric is total number of VHDL lines and another is total cost of VHDL, then the user shall be able to define a metric that is cost per line of VHDL. Thermometers shall be updated at each relevant event.

4.5.7.3   Obtain Status

The process management capability shall provide capabilities for obtaining the status of activities and data objects. It must also provide capabilities for analyzing and rolling up the status into overall project progress indicators.

4.5.8   Adapt Process

The process management capability shall provide capabilities for tailoring the process and plan in real time to account for evolving events. These capabilities are essentially the same as those for "process tailoring" (inserting new sub-processes, changing schedules, modifying product structure, changing roles and responsibilities) as described above, except that they are applied after process execution has begun. Thus they should not require complete replacement of the process

and re-initialization of all states. Furthermore, a history of tailoring changes should be maintained for ongoing analysis.

4.5.9   Response Time

The process management capability shall respond (provide feedback) to all user actions within 2 seconds or less. If an action requires more than 2 seconds, then a progress indicator or appropriate message should be displayed within 2 seconds. Progress indicators are preferred because they let the user know that progress is being made towards completion of the action.

4.5.10   Off-Line Operation

The process management capability shall provide capabilities that allow all tools to work off-line. Here, off-line means that the user's workstation (platform) is not connected to the network or that parts of the network are down. The fact that network reliability is sometimes poor should not prevent work from being performed. This requirement implies that some type of synchronization may be required once communication is re-established.

4.5.11   Other Requirements

Process management include the other requirements that are listed below.

- Security: The capability to provide adequate access control and security.
- Extensible: Scripts are required to add functionality or automate repetitive sequences batch operation
- Scalable: Support is required for small and large projects by incremental addition of HW/SW resources
- Define Reference Process: The process management capability must provide capabilities for capturing, editing, and storing reference processes for reuse.
- Tailor Process: The process management capability must provide capabilities for tailoring reference processes to the specific needs of a project.
- Create Plan: The process management capability must provide features for creating a schedule and allocating resources to the activities of the tailored process.
- Execute Process
- View Process: The process management capability must provide a capability for graphically viewing processes. This includes methods for easy traversal of process hierarchies. This capability is intended to provide performers with on-line access to descriptions of the tailored processes they are enacting.
- Manage Events: The process management capability shall provide capabilities for tracking, signaling, and responding to events during process enactment.
- Evaluate Entry and Exit Criteria: The process management capability shall provide capabilities for defining an activity's entry and exit criteria as a set of expression/action-list pairs such that if one of the expressions evaluate to true, then the associated actions will be fired.

4.6   Architecture

The Process Management Implementation Architecture portrays the functional components of process management within the RDE and indicates how they are interconnected.

4.6.1   Task Manager

The Task Manager is a component of the RDE's Process Enactment capability. The task manager provides day to day task interactions for all users in a distributed fashion, including logging user comments and assisting process compliance.

There is a mechanism for switching between the detailed task view of a user and the process view presented by the Process Browser. Thus, for the Task Manager, a user should be able to ask for a diagram showing where a task fits within the overall process. Likewise, from the Process Browser, a user should be able to ask for the status of all the tasks being performed to complete an activity.

4.6.2   Event Manager

The Event Manager is a component of the RDE's Process Enactment capability. In essence, it is the process execution engine. The Event Manager is responsible for evaluating entry and exit criteria, determining the state of process elements, responding to events, launching actions, and signaling users when events of interest have occurred.

4.6.3   Microsoft Project

For some activities, a complex set of tasks and sub-tasks may need to be planned and tracked (by the performers of the activities). Under such conditions, it would be desirable to let performers construct activity plans using Microsoft Project and export the tasks to the Task Manager for tracking, metrics gathering, and notification.

It may also be desirable to import an indented list of tasks (captured in the Task Manager) to Microsoft Project for scheduling and resource leveling, then pass back the updated information to the Task Manager.

It should be understood, however, that in both of these situations, the function of the interface is to move information between the databases of the RDE and Microsoft Project. From this perspective, the Task Manager and Microsoft Project are simply the tools a user would employ for creating, editing, updating and viewing the data.

4.6.4   Mentor's FlowXpert

The ability to interface with Mentor's FlowXpert needs investigation. It is desirable to interface with FlowXpert in two ways. First, by treating the processes it manages as leaf level activities. Second, by interacting with elements within those activities. With these interfaces, it should be possible to have RDE and FlowXpert work cooperatively in managing a complete development process.

In the first case, entry and exit criteria at the boundaries of a FlowXpert managed process should operate seamlessly between the RDE and FlowXpert. Furthermore, the input and output data objects, plus their states, should also be available to both the RDE and FlowXpert.

In the second case, events that occur within a process managed by FlowXpert should visible outside to the RDE, and events that occur within the RDE should be visible to FlowXpert.

If we were to use Mentor's WFM, we would need a way to dynamically (during process enactment) access the database that it maintains so that status and parameter information could be used outside the WFM by tools on other platforms and other networks.

With what we currently know, the conclusion is that Mentor's WFM cannot satisfy enough of RASSP requirements to make it a long term solution. At most it is a capability we could use in the short term for hands-on learning about WFM usage on real projects.

4.6.5   Other Functional Components of Process Management

Other functional components of process management include are listed below.
- Process Viewer / Browser
- Process Snippet Editor / Manager
- Metrics Viewer / Browser
- Review Manager
- Problem Manager
- Change Manager
- DOORS

5   Data Management

Data Management is one of the major focuses of the RDE. The Data Management services control all product or project data on a project with which the engineer or manager needs to handle. This includes data that is for work in progress, and data that is archived for later use. The product design data includes requirements documents, design documents, hardware schematics, and software source code. The project data includes lists of users and roles, task and schedule information, and metrics. To allow all this information to be managed, the data is controlled with the Product Structure Editor (PSE). This provides the user with a set of functions including Check-In and Check-Out, Design Folder management, Version Control, user access protection, Document State Management, managing Relationships Between Objects and Query capability.

To allow other multiple PSEs or other RDE clients wishing to access database information, an Application Programming Interface (API) has been designed with help from the University of Wisconsin. This interface separates the RDE clients from the database. The clients do not have any database specific code in their executables, they use only the API library. The interface is designed with the requirement that it can be implemented by multiple databases or Data Management Systems. Each database that is used to implement the interface must create a library of subroutines that are defined by the interface. Then when a message is sent using the interface, the

underlying database can execute the request. This allows any desired database to be used as an implementation to the interface. Each company can run the PSE using their favorite database or PDM system, such as SHORE, Sherpa, Oracle, and Metaphase. The pilot implementation has been done using the SHORE object database. This interface was not completed for use at the end of Build 5.

Figure 5 shows the architecture of the database communications. Each box in the diagram is a process, such as a Unix executable program. For Build 5, the clients include the RDE Desktop, PSE, the Source Reuse Utility, and the Peer Review Utility. The Metric Browser and Task Manager are being added as clients. The RDE clients are programs or processes that the user invokes to get at his project data. The clients can be on any platform including Macintoshes and IBM compatible PCs. The client needs only to be on a network (Internet) that can connect with the Communications Server through TCP/IP. The client sends a message to the Communications Server. The Communications Server determines if the message concerns a database that it is hosting, otherwise, it sends the message to a Communications server that hosts the desired database. The databases are distributed across Internet, but can be accessed through any client that has the proper permissions.
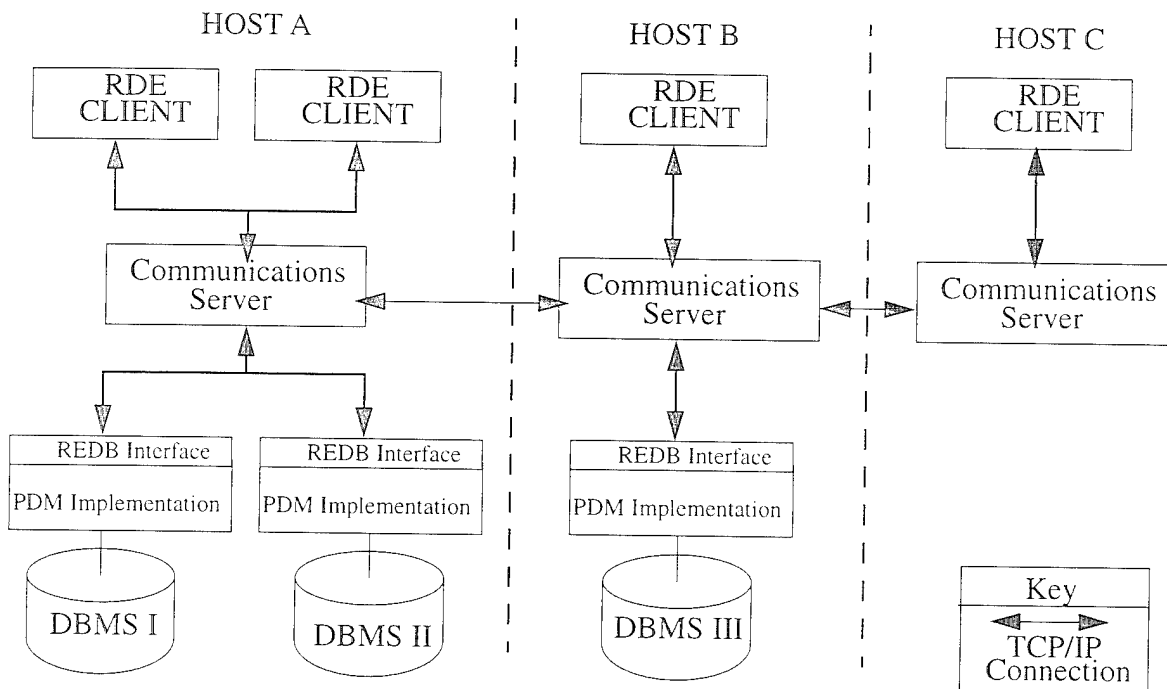

Figure 5 - RDE Database/Communications Architecture

5.1   Distributed Communications

The RDE Database will work across a Wide Area Network. A library of messages that the applications will use are sent to a server. The messages are constructed by the RDE client application in

order to query the distributed database for objects based on object attributes. The following is a set of messages that can made:

- Give me a list of all the child objects of the specified object.
- Give me a list of all the objects that refer to the specified object.
- Copy the specified object into my working directory.
- Give me all the attributes associated with the specified object.
- Give me a list of all objects that match the specified query string.

The messages are sent from the RDE client to the communications server. The client uses the message library to determine the host and port to send the message to from environmental variables that are setup when the tool is installed.

The Communications Server is a background Unix process that mostly waits for a client to make a TCP/IP connection. Once a client connects, the server forks a process that will handle the messages from that client. The server can handle multiple clients at the same time by using the normal scheduling mechanism of the Unix operating system. The client will initially want to connect to a host computer on the same Local Area Network file system to allow good performance. The connection between the client and the server will continue until the server receives a "close" command from the client, or until the socket is broken by some error condition, such as loosing power on the client or server host computer.

The Communication Server can communicate with zero, one, or more databases. Upon starting each database on a particular network, the database process registers with the Communication Server. The Communication Server stores information about the database in a table, and creates a callback that forwards messages destined for the desired database. Most of the messages sent from the client to the Communications Server contain a reference to an object that the client wants to know about. The object name is specified by a URL (Universal Resource Locator) to identify the host and the database containing the object. If the Communication Server does not have the requested database registered, the request for the object is be forwarded to the appropriate server. Each server can be a client of another server. This is server to server communication. The messages that can be sent by an application client can also be sent by a server to another server. The only difference between a server-to-server and a client-server communication is that a client-server stays connected until the application if complete, and a server-to-server communication is only connected for that immediate series of messages.

A Communications Server may not have any databases registered, but may exist only to communicate with other databases. This scheme allows the databases to communicate on a single TCP/IP port. Security software can be used to protect the port from unauthorized access. Each Communications Server that sends a message to a database, sends the same messages with the data formatted in the same format to each database, even though the databases are different. For example, one server knows how to retrieve objects and query Sherpa, and another SHORE.

For those objects that are controlled by a CAD tool, such as a Mentor or Viewlogic design object, the data is stored in the database of the CAD system. Messages, such as how to get the version number of the design, is accomplished by implementing the RASSP Engineering Database Interface (REDI) in terms of the CAD tool interface. In this case the CAD tool would be asked about the version, and passed back through the REDI interface. REDI is a general interface of functions

that can be used to communicate with the databases. The interface is implemented in terms of the interface supplied by the database management system that is storing the objects.

5.2   RASSP Engineering Database

The RASSP Engineering Database contains all design and product data. This includes documentation such as requirements and guidelines, hardware schematics and design documents, software source code, and libraries of software, electronic components, and electronic models, list of personnel and roles on a project, process, scheduling, and task information, project mail, and links to HTTPs on the WWW. The data in the database is stored as objects and can be structured to best organize a project as shown in Figure 6. The objects in a database may have links to objects that physically reside in other databases.

An object in the database contains standard and user defined attributes, and generally represents one or more files on a file system. Each object has a type and a format. For example, an object may be of type 'requirements document' with a format of 'FrameMaker'. The type and format are two attributes of the object. Other attributes currently include: creator, creation date, last modified by, last date modified, version, state, and comment. The user can define his own attributes for his own use.

When file is checked into the database, a database object is created for the file, and the file is stored in a repository or a location on the filesystem where it is protected from the user. Assuming there is not a permission problem, the user can copy out or check out a file. The difference is that checking out a file creates a new version of the object.

Some objects do not directly represent a single file in a repository. One such object is a release object that contains a list of files and versions that can be copied out of the repository for later use. Another object is a schematic that communicates with the CAD tool, such as Mentor, for the required information. In this case, Mentor stores the file in its own repository and the REDB does not attempt to store the Mentor files outside the Mentor environment.

Database objects also have references that are defined by the user or system administrator, references can be defined to relate objects in the database. The referenced object may be in another database and at another geographical location.
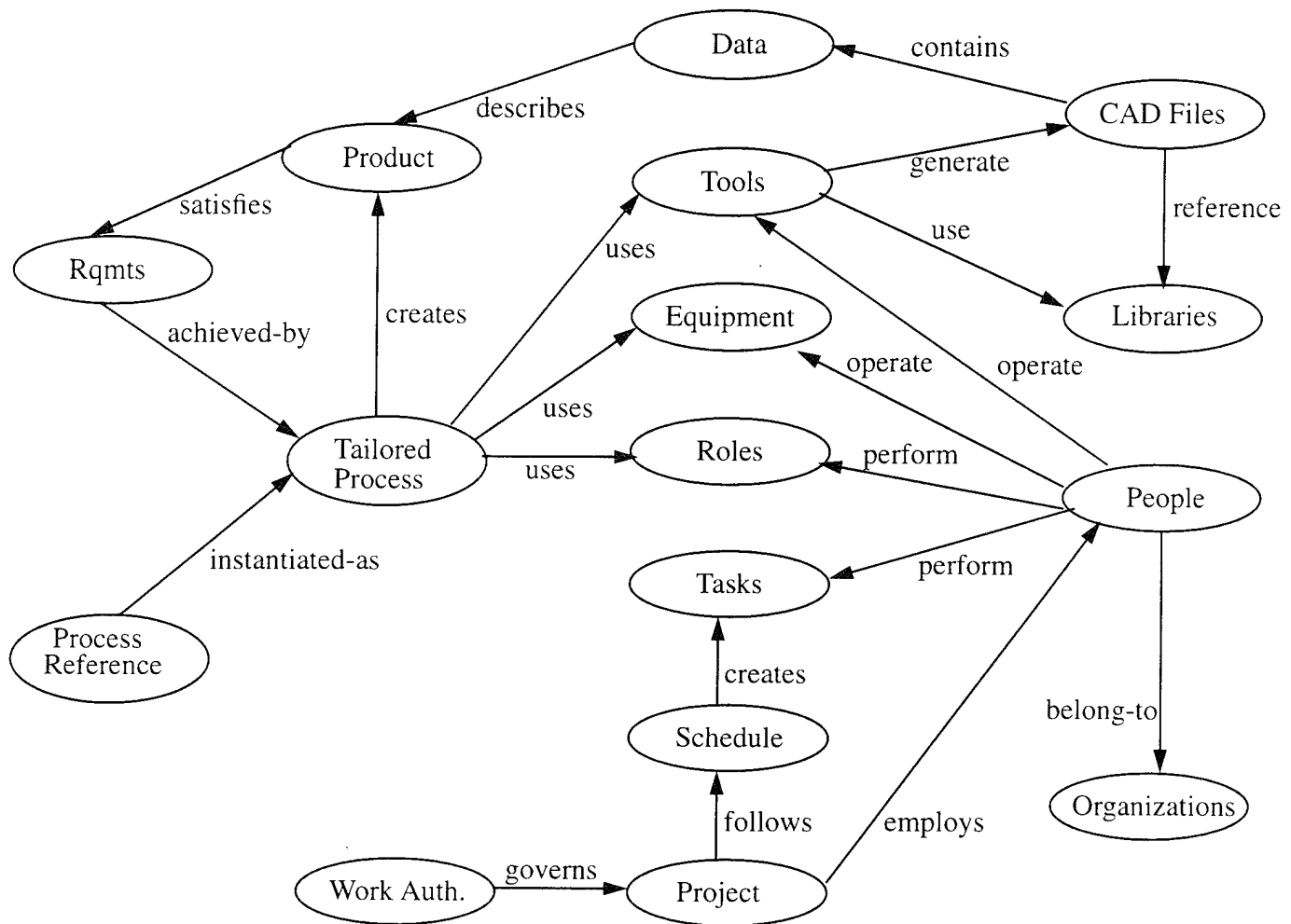
Figure 6 - RDE Information Model

5.2.1   Outside of the Scope the RASSP Engineering Database

This section describes features of the database that would be valuable, but are outside of the scope of the work being performed by the Sanders RASSP team.

We are not implementing a standard interface definition for the product data used within a given user process. It would be great if all data of a certain type had the same format to offer interoperability between tools. Since they do not, we are not forcing the data to be converted to a common format. We will, however, make use of standard commercial translators whereever possible and where it is feasible, create translators.

Though a library object can be created to represent a commercial or company library, we will not attempt to duplicate the information or functions that exist in commercially available tools. Commercial libraries have their own user interfaces and data formats. We will not duplicate the inter-

face or attempt to control and manage the library data, but we do allow a library object to be created that the user can place in his project so that it can to be easily located.

We will not attempt to allow designs to be copied from CAD tools unless it is supported by the CAD tool. Many CAD tools use multiple files and libraries to represent a design. These tools use hardcoded paths within their own data management system and do not provide hooks to copy a design from one system to another.

6   RASSP Network Communications

6.1   Distributed Network Goal

Developing an engineering product across a wide area network such as the Internet leads to a number of data consistency and data sharing problems. The challenge is to provide an environment across the wide area network that imitates the ease of a well-designed local area network.

Currently, data sharing is done in a variety of cumbersome ways. The most common way is through using File Transfer Protocol (FTP). Files are collected into a directory, compressed and moved an intermediate site. The receivers of this data must get the files from the site and move it to their own site. Then the files must be uncompressed before use. This process is cumbersome. The process can causes versioning and configuration management problems, since the users are working outside of a configuration management system.

The goal is to provide engineers with an environment in which they can treat a central repository shared between companies as an extension of their local area network. WIth the file in a central public accessible place, it is easily copied. Let the network be configured so that the transfer across the wide area network is the same as a transfer on the local area network.

This goal is much easier stated than achieved. The primary obstacle to achieving this goal is the fact that most companies place a protective layer between themselves and the Internet. This barrier is commonly referred to as a "firewall". A firewall is a mechanism that is capable of restricting traffic to a network. It is a focal point between two networks for all traffic between those two networks. This traffic can either be allowed or denied by rules programmed into the firewall mechanism. For those without a firewall, a computer on Internet is protected from access by normal Unix security, such as by user names and passwords, and by group and individual permissions set on directories and file.

6.2   Current Network Architecture of the Lockheed RASSP Partners

The picture below depicts the top-level network architecture interconnecting Sanders, Hughes Aircraft, Motorola, and ISX. ISX has no firewall protection. Sanders has an external, unprotected network available for team members to store data. Sanders also has a protected internal network. The Hughes RASSP team works on a network that is behind a firewall, but in parallel with their corporate network. Motorola uses a scheme much like Hughes, except there is an additional connection between the company network and the RASSP network which is also protected by a firewall router.
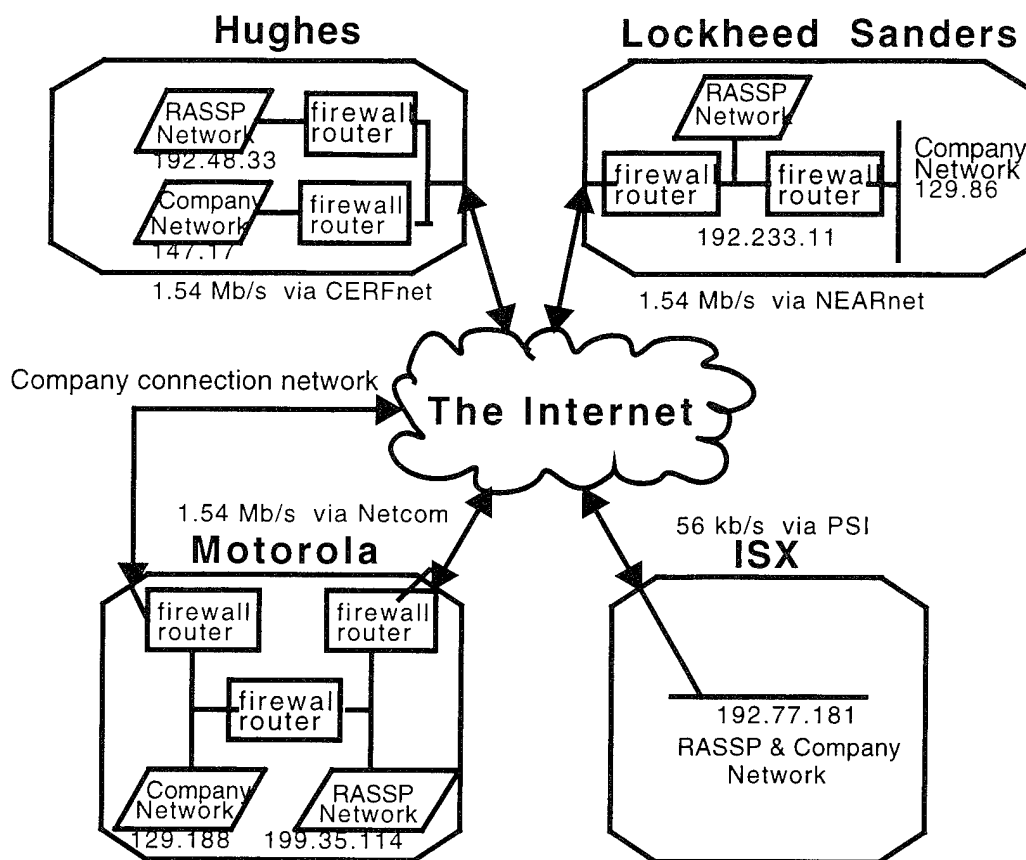
Figure 7 - Physical Topology of RASSP Team Network

6.3   Network Infrastructure Hardware and Software

6.3.1   Commercial Internet Providers

Each of the companies on the RASSP team has chosen a different network service provider, depending on their own need, and the regional variations in pricing and suppliers. Typically, as of May 1995, a full T1 (1.54 Mb/s) Internet connection costs about $40K annually. Lower bandwidths have commensurately lower costs, with 56 Kb/s leased line access typically costing around $18K annually. These commercial Internet providers essentially provide a "spigot" to connect each of the companies local area networks to the Internet.

6.3.2   Firewall Configurations

Since there is no restriction on computer access within the infrastructure of the Internet, it is necessary to protect the local area networks (LANs) from unauthorized access from non-team members on the Internet. For this reason, each of the RASSP team members has implemented some sort of "firewall." between their LAN and the Internet.

At Sanders, this firewall consists of a Cisco router which is capable of packet filtering. The router is configured in such a way as to permit only a restricted set of communication protocols to pass from the LAN to the Internet. Full communication is permitted to the RASSP team members, while only specific protocols in an outbound direction are permitted to non-team member hosts.

6.3.3   User Applications

On top of the underlying RASSP network infrastructure, a suite of applications have been used by the RASSP team to facilitate electronic communication. These applications have addressed messages (via email), documents and design information (via FTP, shared filesystems, and World-Wide-Web), and interactive video conferencing.

Table 2: Sanders Firewall Services

BI-DIRECTIONAL SERVICES	OUTBOUND-ONLY SERVICES	INBOUND-ONLY SERVICES
Domain Name Service	File Transfer Protocol via proxy server	None Permitted
E-Mail	World Wide Web (HTTP) via proxy server	
Internet News	Telnet via proxy server	
Network Time Protocol		

6.3.3.1   Email

Email has been the fundamental communication mechanism in use by the RASSP team. This has been highly successful, with a number of automated distribution lists to permit "broadcast" messages as well as personal correspondence. For interoperability across different software packages and machines, it is essential that mail software take advantage of the MIME (Multipurpose Internet Mail Extensions) standards. As such, the RASSP team has preferred the use of mail packages such as Zmail for Unix machines, and Eudora for the Macintosh or PC.

6.3.3.2   FTP Server

The primary mechanism for sharing documents and design information within the RASSP team thus far has been the use of a central RASSP FTP Server at Sanders. Each of the Integrated Product/Process Development Teams (IPPDTs) is responsible for maintaining an area of the server.

The RASSP team members have had great success using the publicly available "wu-ftpd" FTP server. This server is freely available from Washington University in St. Louis, MO. This server allows for multiple "anonymous" groups to be set up, with the determination of the proper group to be made based upon the place of origin of the FTP request. This permits "anonymous" access to be made by team members, while disallowing any access from non-team members.

6.3.3.3   Shared Filesystems

The ability to utilize shared filesystems across the RASSP network is a relatively new capability, first used by the RASSP Design Environment (RDE) prototype team. This technique allows engineers to easily share common code or design information libraries across the team members. Performance using this mechanism varies with network load, from about 2Kbytes/s to 20Kbytes/s, with typical responses of approximately 5Kbytes/s. This makes it possible for engineers to "check in" or "check out" a typical three or four hundred line VHDL or C++ code file in a few seconds.

6.3.3.4   World-Wide-Web Capabilities

The World-Wide-Web (WWW) is a collection of client/server protocols that provides easy access to information from client software available for Unix machines, Macintoshes, and PCs. The primary advantage of WWW client software is that it is relatively easy for a novice to gain access to a wealth of information.

Within the RASSP program, we have been examining ways for using this technology within the team. Notable uses of WWW services have included a distributed address book (rolodex), document server, calendar, team glossary and acronyms, and metrics reporting.

In order to view WWW information, a WWW browser is required. The preferred browser on the RASSP team is "Netscape Navigator", by Netscape Communications Corporation. A variety of WWW servers are being used by the RASSP team members, including the freely available National Center for Supercomputing Applications (NCSA) HyperText Transfer Protocol Daemon (HTTPD), the freely available CERN HTTPD, and the commercial Netscape Server.

6.3.3.5   VideoConferencing

Electronic videoconferencing has been used within the RASSP team to send and receive audio and video data, primarily for conducting distributed meetings and reviews. The preferred product on the team, due to its low cost, is the CU-SeeMe software from Cornell University. Recent versions of this product support integrated audio and video for both the Macintosh and PC platforms, with multi-way conferencing support.

Within the RASSP team, the CU-SeeMe software has been used exclusively on Macintosh computers. The only additional hardware requirement beyond an AV-compatible Macintosh is to have a video camera. The two tested products by the RASSP team have been Connectix' "QuickCam," and VideoLab's "FlexCam". The QuickCam is an inexpensive ($99) black and white camera giving 320x240 resolution, while the FlexCam is more expensive ($499) color camera providing 640x480 resolution. Both cameras perform adequately with the CU-SeeMe software, while the FlexCam provides increased capabilities for other applications.

6.4   An Internet NFS Server

In order to efficiently share data, a way must be devised so that company A can easily place data in a place where company B can easily retrieve it. A scheme using tar, compress, and ftp does not meet that goal for the purposes of this discussion. The goal is to make this wide area data transfer as seamless as a local area network data transfer. That is, make it so a user can copy data from his local directory to another directory, which may be local or across the country, using the exact same procedure.

The plan to do this is to engineer a secure Network File System server, secure the communications between this host and remote hosts at the various companies, and create an NFS filesystem across the Internet. This would allow a person at company A's secure host to copy a file from his host to the remote NFS server host.

6.5   How to Create an NFS Server

The first step to creating an NFS server is to configure a Unix machine with only the NFS service available. Edit the configuration files (inetd.conf) to ensure that no other network services are available. Then remove the binaries for those services from the operating system. Some of these services are HTTP, telnet, rlogin, rsh, finger, talk, and write. There are many others.

Once the machine can be considered a bare-bones NFS server, then the security must be provided for the NFS transmissions. We use the NetLock software. This software takes an IP packet, completely encapsulates inside a packet generated by NetLock, and transmits and receives the new packet on NetLock specific ports. A firewall can be configured to allow NetLock traffic to pass through since the ports NetLock uses are well known. Presumably, the IP addresses of any hosts wishing to communicate to the NFS server are also well known.

The NetLock software performs the authentication with other NetLock hosts. The hosts can be configured so that the only allowed communications are with other NetLock hosts.

Our proposed NFS server configuration entailed the NFS server "straddling" the existing corporate firewall. This would be accomplished by configuring a router as an NFS server firewall. This router would only allow NetLock traffic to pass in a bi-directional fashion. No other traffic is allowed through the firewall.

Once this firewall has been tested, a second ethernet card can be added to the NFS server to provide for a connection to the internal corporate network. It must be realized that this alternate path to the external network is actually much more restrictive than the existing path.

The system can also be configured so that the existing firewall can allow Netlock traffic to and from the internal corporate network. This would alleviate the need for a second router and a second ethernet card. There would only be one path between the external and internal networks. The existing firewall would have to be relaxed to allow NetLock traffic to pass through it.

Once the NFS server has been completely configured, a program is run to detect changes to specified files in the file system. We chose the Tripwire program. This program uses multiple "checksum" procedures on specified files to ensure that the machine has not been compromised in some way. These checksums are compared against an original database that is maintained on a physically protected read-only media (such as a hard disk or floppy disk that has been physically configured to be read-only.) Tripwire reports all changes between the original database and the current configuration. When set up to run daily, Tripwire provides a good measure of intrusion detection.

7   Engineering Environments

The RASSP Demonstration team and first Benchmark team have used VHDL and Ada for engineering of their projects. This section describes the tools that were used in the VHDL and Ada engineering environments.

7.1   VHDL Tool Environment

RASSP activities that have been completed in VHDL have been on the Demonstration, and the Benchmark 1. Both of these environments are similar, but the Demonstration is further along due to its greater scope. The tools used for VHDL development include emacs as a text editor to create the behavioral VHDL source code. The source code is then analyzed, elaborated, simulated, and debugged using the Vantage Spreadsheet VHDL Development Tool. The source code is controlled with the Unix SCCS (Source Code Control System), for engineers to check in and check out working and released versions of the VHDL source code. After the simulation is successfully completed, Synopsis is used for FPGA gate level synthesis. Mentor's Quicksim II is used to add timing before Board Station is used for circuit board layout, completing the VHDL design.

7.2   Ada Tool Environment

The RASSP activities that have been completed in Ada have been on the Demonstration, and the Benchmark 1.

The Benchmark Environment uses Ada for work on a 68040 processor. There are 2,000 to 3,000 lines-of-code that are being developed to control the benchmark hardware. The software development platform is a Sun workstation using SunOS 4.1.3. The top level software is designed using Cadre Teamwork. This tool represents the design with data flow diagrams that describes the functions and the data that are passed between the functions. After the design is completed, the software is written using the emacs editor in Ada mode. This mode allows the indentation to automatically conform to Ada conventions. The source code is controlled using SCCS, allowing the engineers to control and track the versions of the software.

The software is compiled on the Sun with the Alsys cross-compiler on the Sun that produces 680x0 object files. The object code is then linked with a Microtech linker that allows a simulation to be performed on the Sun workstation. The software is then debugged and tested using the simulator.

After the initial software testing, the object files are linked into the Styx Interface where the software can be executed along with the VHDL simulation from Vantage. The co-simulation of Ada with VHDL allows software that controls the hardware to be tested. The entire system can be tested in this manner.

The demonstration team environment is similar to the benchmark environment. The software was developed on a Sun workstation. The top level design was done using Cadre Teamwork. The software was written in a text editor, cross compiled with a Mercury compiler targeted for the i860XR microprocessor, and debugged with the Mercury adagdb860 debugger. After the executable image was generated, the coff860 tool generates "Common Object File Format" (.coff) files that are loaded into the VHDL simulation and run on the virtual prototype. This allowed the software to interact with and be tested with the hardware described in VHDL.

No automatic code generation was used for the Ada development. Cadre Teamwork provides the capability to generate Ada code from the data flow diagrams. The engineers decided that the information that was required to be entered into this tool to allow it to produce useful code would not be as productive as typing the information using a text editor. Code generation is an area that can be improved upon to increase the productivity in developing software.

There is currently not much reuse, either from previous design diagrams, or code libraries. The compilers come with board level support libraries that contain operating system routines for the processors. No other commercial libraries, public domain, or in-house libraries were used for these programs. This is an area that can be improved.

8   Appendix A - RDE User's Guide

Rapid Prototyping of Application Specific Signal Processors (RASSP)

**RDE
System
Description**

**APPENDIX
A**
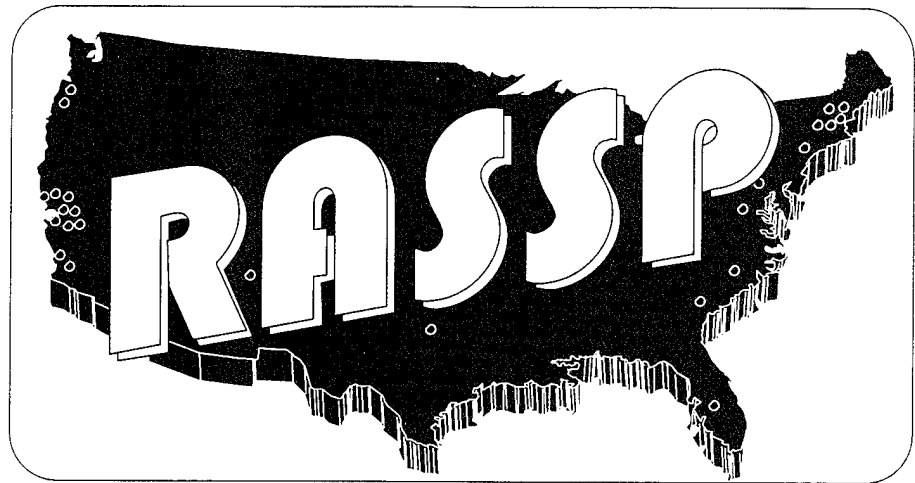
22 December 1995

**Advanced Research Projects Agency
Naval Research Laboratory**

SANDERS

A Lockheed Martin Company

Document #AVY-L-R-00250-101-A

TABLE OF CONTENTS

LIST OF FIGURES

Glossary of Acronyms

AI	Action Items
CWD	Current Working Directory
DMAT	Data Management Access Tool
ggnats	Problem Report Tool
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocal
IPD	Integrated Product Development
PC	Personal Computer
RASSP	Rapid Prototyping of Application Specific Signal Processors
REDB	RASSP Engineering Database
RDE	RASSP Design Environment
TDM	Team Design Manager
WWW	World Wide Web

Chapter 1: Getting Started

Chapter Content

Refer to this Chapter for:

- Conventions
- User's Guide Overview
- Introduction to the RDE
- Using the RDE
- System Requirements
- Glossary of Acronyms
- Customer Support

Conventions

The following conventions will be used throughout this document:

- Acronyms are used for common or frequently used terms, refer to Chapter 1: "Glossary of Acronyms" on page 17.

- Information that may be typed in by the user, and information being displayed from files will be displayed in 10 point Courier, usually on a separate line, as follows:

```
cout << "The RDE is great...Want some ice?" << endl
```

- Basic knowledge of Unix and X-Windows is assumed.

User's Guide Overview

This User's Guide is part of the RDE Release 1.0 distribution. It consists of nine chapters that provide the information and instructions to guide the RDE user through the process of developing a RASSP-based system, i.e. creating and implementing electronic designs in an integrated environment to facilitate an efficient concurrent engineering process.

The following list outlines the contents of this guide.

° Chapter 1: Getting Started
° Chapter 2: The RDE Desktop Utility
° Chapter 3: The Product Structure Editor
° Chapter 4: The Remote Data Access Utility
° Chapter 5: The Log Utility
° Chapter 6: The Multi-Metric Browser Utility
° Chapter 7: The Problem Report Utility
° Chapter 8: The Reuse Utility
° Chapter 9: The Review Utility
° Chapter 10: The Task Manager

Getting Started

This "Getting Started" Chapter provides a Chapter summary of the User's Guide. The information stated in the "About This User's Guide" Section is reiterated and expanded. Getting Started includes: Conventions, a basic Introduction to Using the RDE, System Requirements, Acronyms, and important Customer Support information.

The RDE Desktop Utility

"The RDE Desktop Utility" Chapter provides a guide to the functionality of the RDE Desktop strip menubar and menu items, and how to invoke RDE Utilities and Services from the Desktop.

Chapter coverage includes: Invoking and Exiting the RDE Desktop, Navigating the RDE, Tool Launching and Encapsulation, Help.

> *NOTE: This manual assumes you're familiar with basic Unix and Motif terminology and operations. Refer to appropriate documentation as needed.*

The Product Structure Editor

"The Product Structure Editor" Chapter provides a guide to the functionality of the Product Structure Editor (PSE) and how it is used to manage data for an engineering project.

The Remote Data Access Utility

"The Remote Data Access Utility" Chapter provides a guide to the functionality of the Remote Data Access Utility (RDA) and how to access remote configuration management controlled data repositories which house the distributed RDE database. This multi-platform client currently runs on Macintosh, Unix, and Windows platforms.

The Log Utility

"The Log Utility" Chapter provides a guide to the functionality of the Log Utility and how to log and access data for use in a project. The ability to make high and low level annotations is supported by the free format of this utility.

The logged data is automatically stored in the RDE database within the project context. The Log Utility provides a query engine for log retrieval.

The Multi-Metric Browser Utility

"The Multi-Metric Browser Utility" Chapter provides a guide to the functionality of the Multi-Metric Browser Utility and how to access and display metric data. The RDE automates data collection and metric generation as much as possible.

Metrics will be customizable to provide any Integrated Product/Process Development Team (IPPDT) member with useful metric data. The Multi-Metric Browser retrieves data from the RDE database as requested to populate the desired user form or report.

The Problem Report Utility

"The Problem Report Utility" Chapter provides a guide to the functionality of the Problem Report Utility and how to define and submit problem reports for use by the Problem Management System. The complete Problem Management System facilitates the handling of problems throughout the product development life-cycle. Remote access to the Problem Management System facilitates problem management for the distributed development team and the Virtual Corporation.

A key challenge of development teams is to effectively identify, contain, and resolve development problems. To facilitate and automate this critical need the RDE uses the GNATS Problem Management System. The Problem Report Utility provides a GUI problem submission utility as a replacement for the GNATS command-line send-pr. The Problem Management System automatically inform responsible parties and sends appropriate notifications. Problem Report data is available for metrics collection and display.

The Reuse Utility

"The Reuse Utility" Chapter provides a guide to the functionality of the Reuse Utility and how to locate and acquire data for reuse in another project. Users may add to, query, and update the reuse database.

Extended reuse component attributes; e.g. common keywords, logical keywords, descriptions, units, interfaces; encourage the use of this utility and simplify the documentation of and the access to reusable development components. A variety of reuse library management capabilities facilitate the use of this utility: import, export, backup, deletion, redundancy, metrics display. Remote database synchronization facilitates reuse for the distributed development team and the Virtual Corporation.

The Review Utility

"The Review Utility" Chapter provides a guide to the functionality of the Review Utility and how to perform reviews in a project. The Review Utility supports and automates the entire review process. Review data is encapsulated in a Review Package which is maintained throughout the review process.

Performing consistent accurate Technical Peer Reviews help ensure compliance with the development process and a quality product. The Review Utility facilitates on-line review by a geographically diverse team of reviewers. The Review Utility data is available for metrics collection and display.

The Task Manager

"The Task Manager" Chapter provides a guide to the functionality of the Task Manager and how it is used to control tasks on an engineering program.

Introduction to the RDE

The RDE facilitates Integrated Product Development (IPD) by providing a collaborative work environment. The RDE provides support for automating the product development process to improve product development, specifically with respect to cycle time, product cost, and product quality. The RDE enables the IPD philosophy with its support of rapid iterations, incremental promotion, and scalable configuration management controls. The IPD approach can be employed during all phases of a product's life cycle from conceptual and detailed design through production to field support, refer to Figure 1 on page 2 of Chapter 2 for Product Development Maturity Matrix.

A wide variety of disciplines will be utilized throughout the product development process which requires the use of many classes of tools including tools for program and project management, requirements capture and analysis, algorithm development, software engineering, and electrical and mechanical hardware design, modeling, and simulation.

The RDE provides technologies that fully support concurrent design, development, and the electronic exchange of information, e.g. status, schedules, and product data. A fundamental thread in supporting IPD is communication; the RDE commercially available high-speed communication services allow for linking to geographically diverse sites. Since team members represent different companies, organizations and product development disciplines; the RDE supports a heterogeneous computing environment, links between geographically diverse locations, tolerable configuration management, diverse product data and diverse tools. The RDE supports an improved product development process allowing for rapid iterations, incremental promotion, and scalable configuration management controls.

Using the RDE

The RDE must be installed on compatible system(s), for detailed installation information and system requirements refer to the Administrator's Guide (also refer to Chapter 1: "System Requirements" on page 16).

Starting the RDE

When the RDE is properly installed a user may invoke it via the Unix command-line by typing:

```
rde<ENTER>
```

This will display the RDE Desktop Utility. Now you have access to the RDE Services and Utilities.

Understanding the RDE

The RASSP Proliferation Team offers RDE training courses. If training in the use of the RDE and the RASSP Process is desired refer to Chapter 1: "Customer Support" on page 18.

This guide is intended for use as a reference and not a tutorial. Users of this material should be familiar with RASSP and the RDE. Users also must be literate in Unix and X-Windows.

System Requirements

This section describes the platform. hardware and software requirements for the RDE. This includes memory and disk space requirements. Use this information to make sure you meet the minimum hardware and software requirements as they are crucial for proper installation and operation of the RDE software.

Platform Requirements

The RDE software currently runs under Sun SPARC platforms running SunOS version 4.1.3 only.

Hardware Requirements

To accommodate the RDE, the following memory resources are required:

- 50 Megabytes Hard Disk space

- 16 Megabytes RAM

Software Requirements

The RDE software is dependent on the following software products for proper functioning:

- X-Windows System, Version: X11R5, or X11R6.

- TDM (Team Design Manager) from Cadence is required for configuration control of development data.

NOTE: Full RDE functionality and services are provided without the explicit use of a Configuration Management (CM) or Version Control (VC) System. However, the use of some kind of CM / VC System is encouraged. Many CM / VC System benefits are noted in the development process.

- GNATS by GNU is used as the Problem Management System. This is public domain software that may be obtained via ftp from prep.ai.mit.edu:/pub/gnu/gnats*.

NOTE: GNATS is not required; however, some Problem Management System must be in place. The product development life cycle may be severely hampered by the absence of a Problem Management System.

Glossary of Acronyms

AI	Action Items
CM	Configuration Management
COTS	Commercial Off-The-Shelf
CWD	Current Working Directory
GGNATS	GUI GNATS
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IPD	Integrated Product Development
IPPDT	Integrated Product/Process Development Team
PC	Personal Computer
RASSP	Rapid Prototyping of Application Specific Signal Processors
RDA	Remote Data Access Utility
RDE	RASSP Design Environment
REDB	RASSP Engineering Database
TDM	Team Design Manager
VC	Version Control
WWW	World Wide Web

Chapter 2: The RDE Desktop Utility

Chapter Content

Refer to this Chapter for:

* Introduction
* Starting the RDE
* File Menu
* Set Working Directory
* Edit Preferences
* Edit Categories
* Show All Tools
* Show Function Names Show Tool Names
* Install New Tool
* Quit
* Menu Bar Categories
* Using a Tool Palette
* Customizing a Tool
* Saving Tool Information
* Launching Tools
* Help Menu

Introduction

The RDE is an environment shell in which working conditions can be customized and tools can be accessed and launched. Categories of tools can be set up in separate windows (tool palettes) so that there is easy access to launching of the tools that are used most. The way that the categories are set up is remembered from session to session - in regards to the categories that are defined, which tools are in those categories, and the position and size of those category tool palettes. The tool palettes contain icons for the tools - providing quick and easy recognition and execution access.

GENERAL NOTES

* Use the TAB key or point and click with the mouse to move between the fields of a window.

* Selecting the Dismiss button displayed at the bottom of windows will close the current window.

Starting the RDE

Type rde in an xterm window and press Return.

> *NOTE: If you receive any error messages (e.g., connection re-*
> *fused) after typing "rde," the database server may be down.*
> *Contact the System Administrator (SysAdmin), to start it up*
> *again.*

After a few moments, the RDE menu bar is displayed.
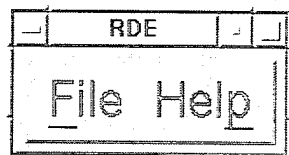
FIGURE 1.   RDE Menu Bar

This menu is the entry way to all the functionality of the RDE. By de-
fault, it contains the *"File"* and *"Help"* choices, and can be custom-
ized to include more choices - namely a choice for each category of
tools that are used. To add categories to the menu, see Figure 13,
"Show All Tools (Tool Database)," on page 29.

File Menu

The File menu allows users to set the current working directory, edit
user preferences, show all the tools available in the database, toggle
between the tool names and the function names for the tool icons, in-
stall a new tool into the database, and quit from the RDE.
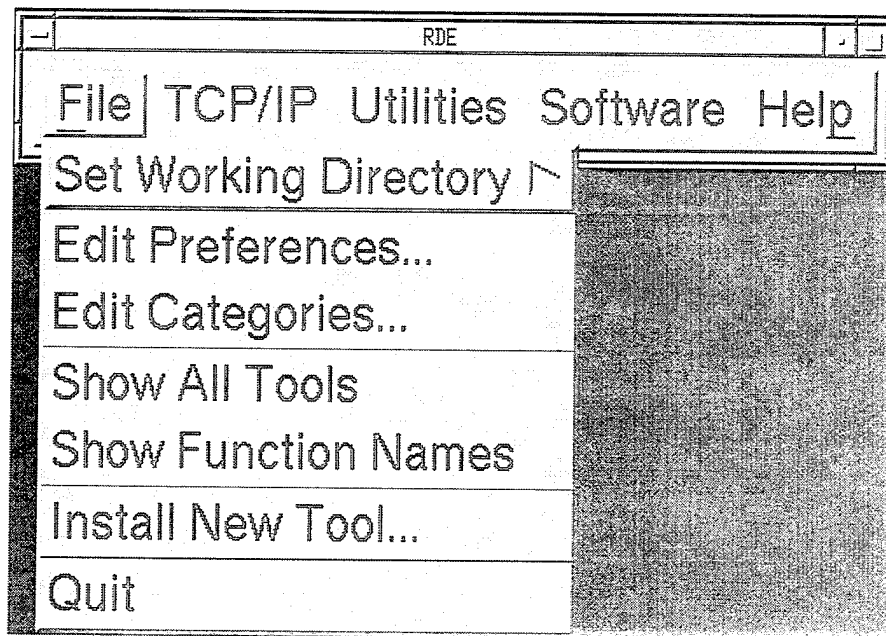
FIGURE 2.   File Menu

Set Working Directory   There are three methods of setting the working directory: Choose
from file browser; Choose from history list; and Choose from alias
list.
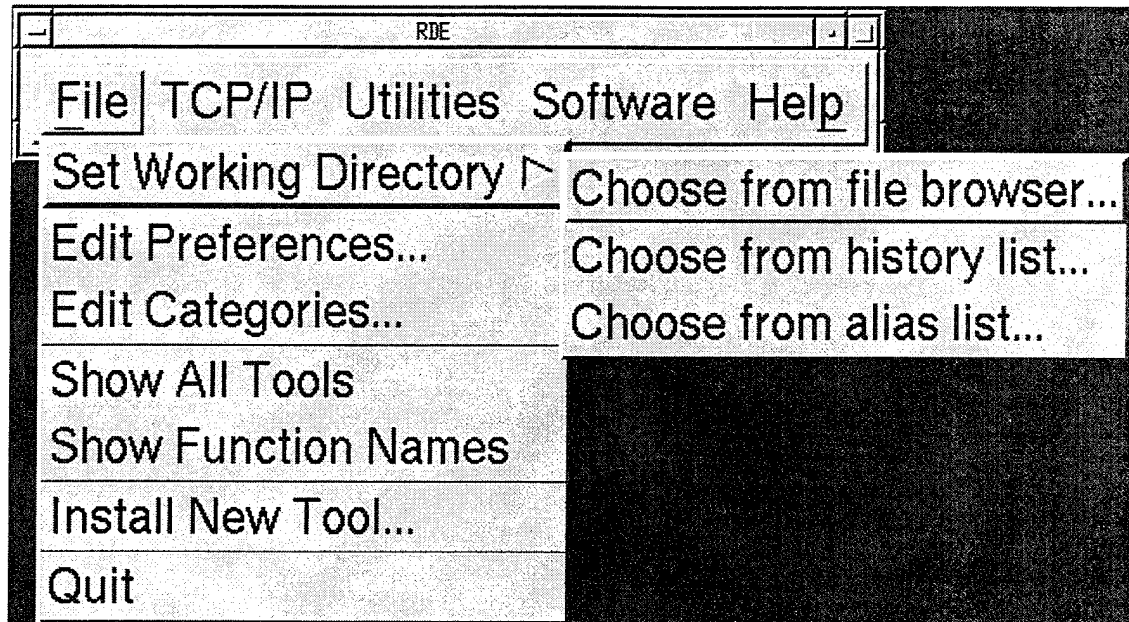
FIGURE 3.   Set Working Directory

Use the Set Working Directory menu to establish from where you
want to launch a tool. If the tools are launched through scripts, and
the scripts do not change directories prior to actually running the ex-
ecutable, the Current Working Directory (CWD), as set by these ac-
tions, will be the place from where all tools are launched. This
implies that as the tool is running, it will consider the current directo-
ry the one that was chosen through Set Working Directory.

When you select the CWD, the RDE will place all files created (by
any tool) into the specified directory until you specify a different
working directory through the tool, or unless the tool explicitly spec-
ifies a different directory on its own. The very first CWD is the one
in which the RDE is launched; and then the CWD is remembered
from session to session. The CWD can be changed by the user at any
time.

Choose from the File Browser

Filter box —————

Directories list box —————

Selection box —————

File Selector
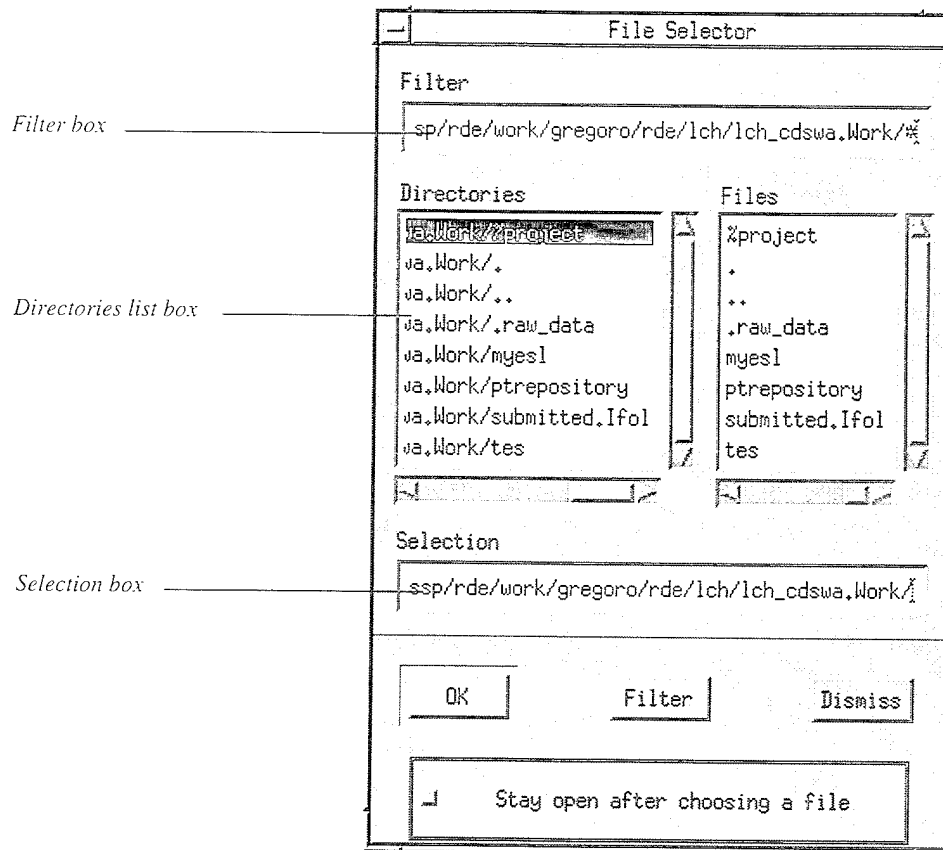
FIGURE 4.   Choose from File Browser

The *"File Browser"* will most likely be used when the user wants to
define an alias, and not have to type in the full path of the directory.
Setting the CWD through this method, and then using the alias dialog,
mentioned below, to make an alias for the CWD (by hitting the "Add
cwd" button), the user will have avoided typing in the entire path to
the directory.

○   **Filter** - Enables filtering using information provided in the Fil-
ter textfield. The default for the Filter is to display all directories
in the directory. When the window is first displayed, you are
shown directories at the level from where you either launch the
RDE or last update the working directory (from this window).

- **Selection box** - The path is displayed in the Selection box. Verify that the path is where you want files to be placed when they are created within the RDE.

- **Directories** - Use the directories list to locate the directory where you want newly created files to be placed. As you traverse the directory, the path name is automatically constructed and displayed in the Selection box.

- **Stay Open -** Toggle to control keeping the window open after choosing CWD.

- **OK -** Save the path, as the CWD, and close this window.

Choose from the History List

| DirectoryHistoryDialog |
| Directory History | Maximum size: 7 |
| /local/projects/rassp/rde/ESL/lch/sun/save/ |
| /auto/sungod10/homes/dsp/gregoro/ESL/etc/ |

__| Stay open | OK | Reset history length | Dismiss
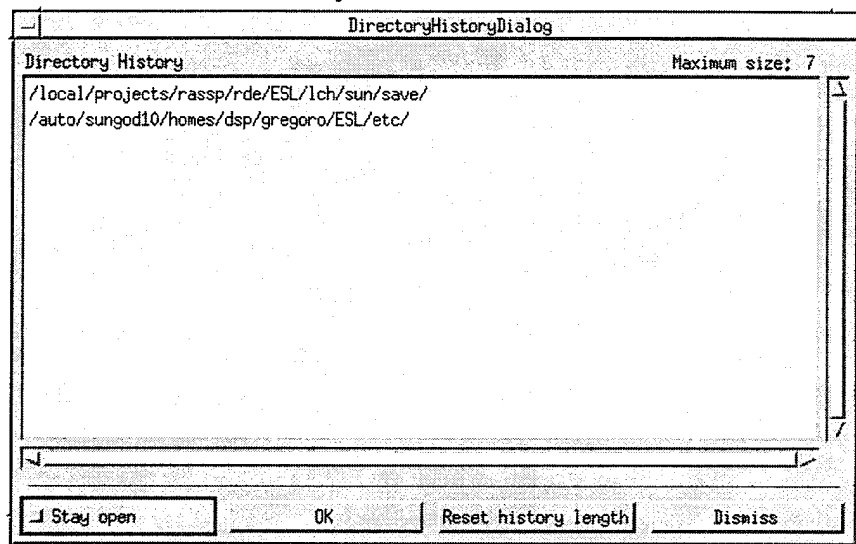
FIGURE 5.   Choose from History List

The *"Directory History"* Figure 5 on page 23 is mainly used for setting the CWD and seeing the last few directories that were worked in. The number of directory names it keeps track of, is set by the user.

- **Directory History** - When the window is first displayed, you are shown directories that have been previously selected from the File Selector window. These directories are saved in the .rdetools file in the user's home directory - for retrieval upon each start-up. A single click followed by hitting the OK button will automatically dismiss the window.

- **Stay Open** - Toggles window feature to stay open after choosing CWD

○ OK - Set the selected directory as the CWD, then closeness.

○ **History Length** - The length of the list is controlled by the user. The current max value of the length of the list is shown in the upper right corner and can be reset by hitting the 'Reset history length' button.

○ **Dismiss Button** - This button will close the window no matter if the Stay open toggle is on or off.

CHOOSE FROM ALIAS LIST

Hitting "Expand" widens the window
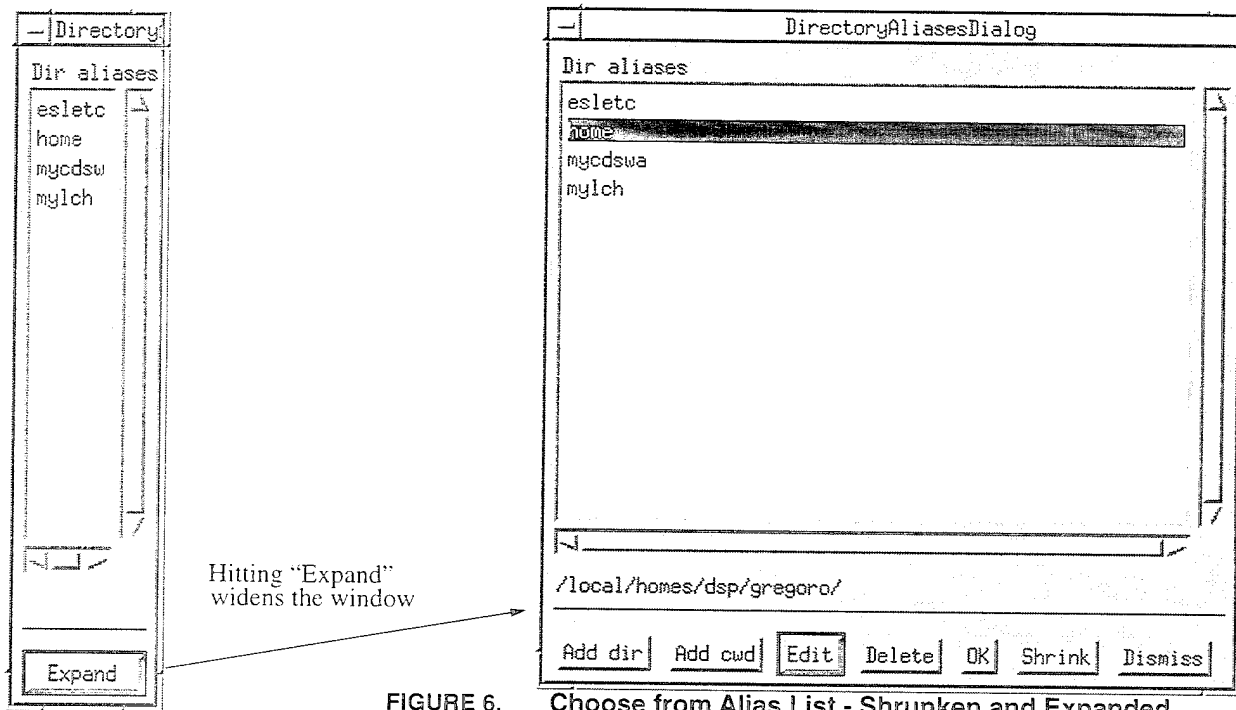
FIGURE 6. **Choose from Alias List - Shrunken and Expanded**

This *"Directory Alias"* Dialog is intended to remain on the desktop at all times, and take up very little space while doing so. Its attractive feature is that it can be shrunk and expanded. In the shrunken state, it gives just enough view of the alias names so that they are recognizable and selectable. The window is almost always kept shrunken. When aliases need to be edited, added, or deleted, then the window would be expanded for making the changes. A simple single click on the alias in the list will immediately change the CWD (updating the history list simultaneously). Typically, the user clicks on a desired alias immediately before launching a tool - so that the tool considers the selected directory its CWD.

- **Expand** - Hitting the Expand button widens the window to show additional commands for aliasing.

- **Add dir** - Add a new directory alias to the list.

- **Add cwd** - Add whatever the new cwd is to the list and give it an alias.

- **Edit** - Edit the highlighted alias.

- **Delete** - Remove the alias from the list.

- **OK** - Make the highlighted alias the cwd, and close the window.

Figure 7 & 8 show some of the dialogs that affect the user's alias list.
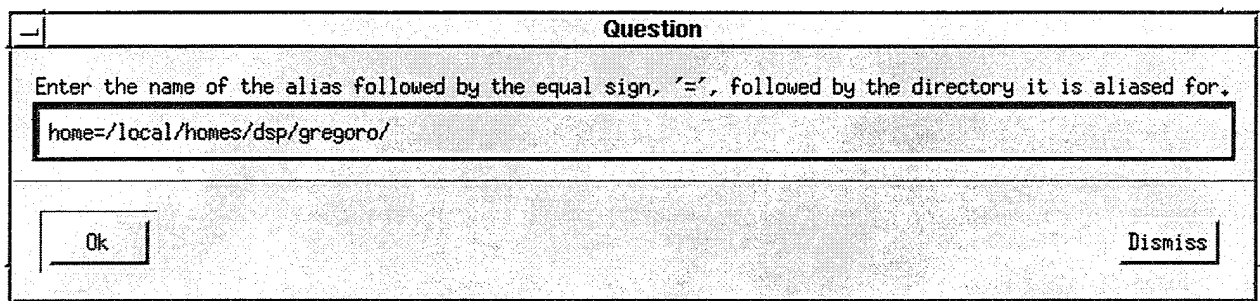
FIGURE 7.   Add Dir Window

- **Add Dir** - This window allows the user to make an entry to the list.
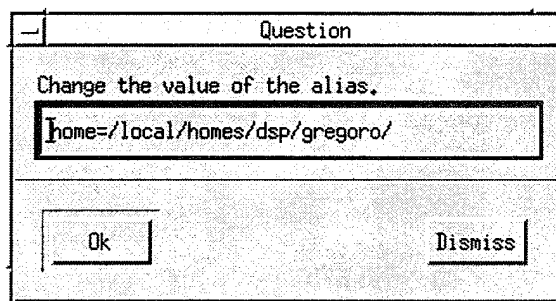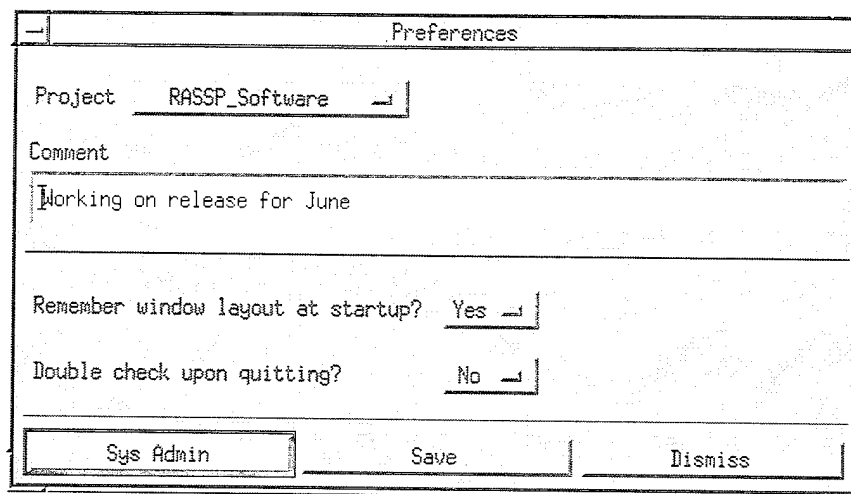
FIGURE 8.   Edit Dir Window

- **Edit Dir** - This window allows the user to change an entry in the list.

Edit Preferences

Use the *"Edit Preferences"* dialog to set user preferences, to set the project, with a comment (for the current working rde session), and the SysAdmin directory data. The preferences, project and comment will be remembered from session to session.

FIGURE 9.   Edit Preferences

The ToolTalk service, if activated, will communicate through the RdeBulletinBoard (a hidden background server), to other applications using the same list of projects, the new project chosen. When a different project is selected as the current project, then the other applications will be notified via the RdeBulletinBoard, so that they can update their current project data.

○   **Project** - Select a new project from this list. This list is defined in a file called ".rdeprojects" in the $RDE_HOME/etc/ directory.

○   **Comment** - Enter a comment that briefly describes what you're doing on the project.

○   **Remember Layout at Start-up** - Remembers the layout, size and position on screen of the fixed rde desktop application windows, and NOT any windows that come from tools launched from the rde desktop.

○   **Double Check Upon Quitting** - Asks the user if he really wants to quit, after choosing *'Quit'* from the File menu. If the user does not want the double check option turned on, then this should be set to 'No'.

- **Save** - Save the preferences to the user's configuration file immediately.

- **Sys Admin** - Bring up the window to set information known to the rde system administrator. See Figure 10, "Set Tool Dir Info Window," on page 27.

The Set Tool Dir Info Window provides the ability for a user to determine the location of the tool_log file, rde dynamic library, and the real dynamic library. The rde dynamic library is intended for use with tools that are declared to be Dynamic when they are installed, and the real dynamic library is used along with it. The main purpose of the rde dynamic library is to try to collect as many metrics about the tool as it is running as possible. The dynamic library capability is currently disabled.

Initial Database Setup	
Tool log dir:	/local/projects/rassp/rde/build4/tool_log
Desktop dynamic lib dir:	/local/projects/rassp/rde/build4/etc
Real dynamic lib dir:	/usr/lib/
Okay	Help	Dismiss
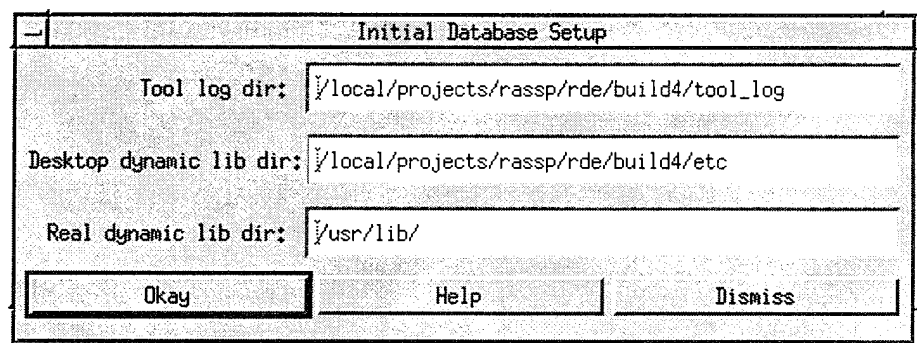
FIGURE 10.   Set Tool Dir Info Window

- **Okay** - Sets the fields information for subsequent use.

- **Help** - Pops up a window with a short description of each field to be entered. Figure 10, "Set Tool Dir Info Window," on page 27.

```
┌─────────────────────────────────────────────────────────────────┐
│ ─                          Message                                │
├─────────────────────────────────────────────────────────────────┤
│   Tool log directory:   Place where parent will put tool metrics. │
│ s Desktop dyn lib dir:  Location of the libc.so.xx.xx that was made for this │
│ ⃥                       desktop ($RDE_HOME/etc/libc.so.1.99)       │
│   Real dyn lib dir:     Location of the system's libc.so.xx.x dynamic library. │
├─────────────────────────────────────────────────────────────────┤
│                          ┌──────┐                                 │
│                          │  OK  │                                 │
│                          └──────┘                                 │
└─────────────────────────────────────────────────────────────────┘
```
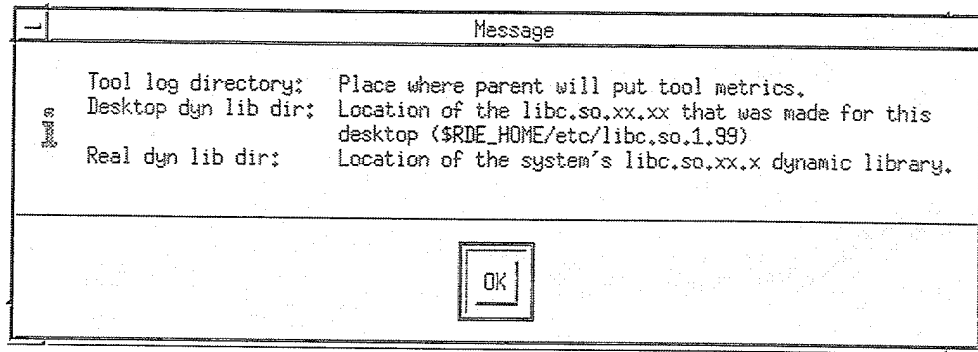
FIGURE 11.   Set Tool Dir Info Help Window

Edit Categories

Select "Edit Categories..." from the file menu. The "Category Editor" Figure 12 on page 28 allows the user to add and remove categories with ease.

After selection 'Remove' becomes active.
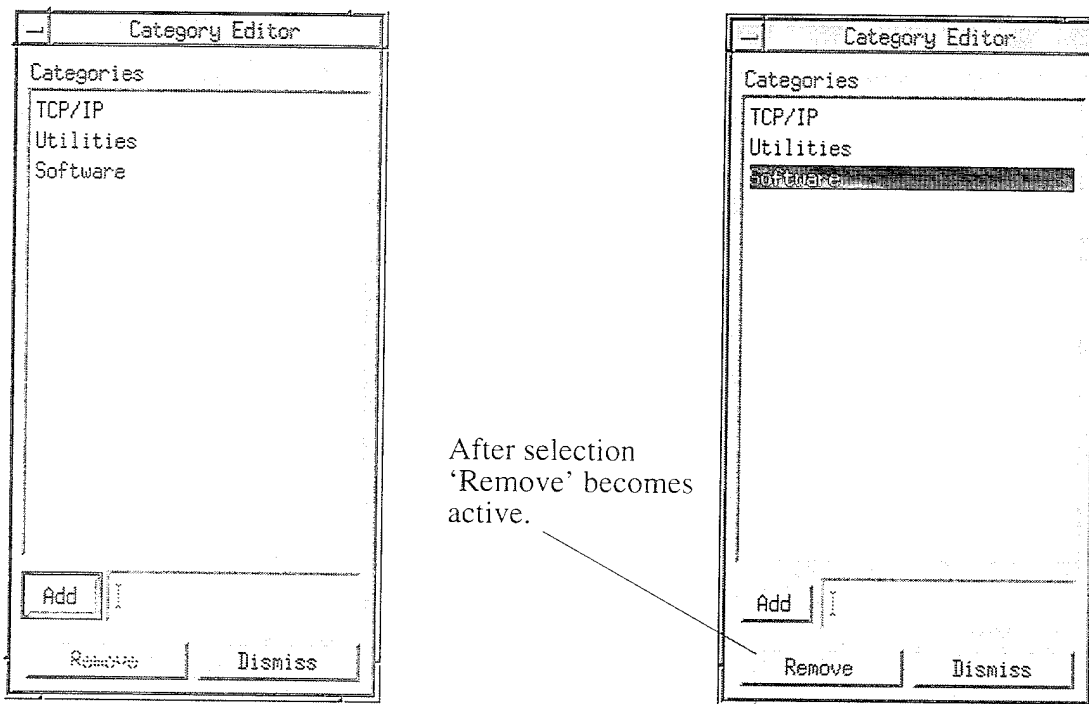
FIGURE 12.   Category Editor Window-before and after selection.

- ° **Add** - Make a new category and add its name to this list and to the RDE menubar. Either hitting the return key after typing in the name, or hitting this button, will add the category.

- **Remove** - (Becomes active only after a selection is made.) Remove the category by closing all the open tool palettes for that category and putting all its tools back into the tool database window. Remove its name from this list and from the menubar.

Show All Tools

Use the Show All Tools menu to show all the tools that reside in the database. (see Figure 13, "Show All Tools (Tool Database)," on page 29). These are all of the tools that are available to the user. These tools can be launched straight from this window, or launched from category tool palettes. Double-clicking on the icon, will launch the tool.

To make a category tool palette, click on the *"Make new category"* button at the bottom of the dialog. (For more information see Figure 14 on page 30)
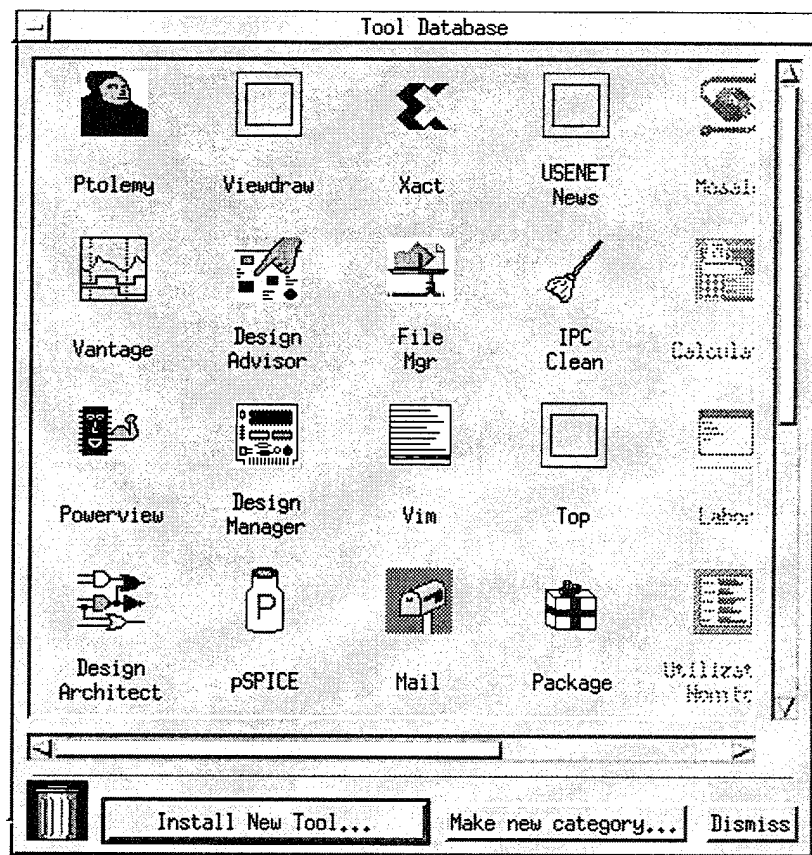
FIGURE 13.   Show All Tools (Tool Database)

   ◦   Install New Tool - Allows the user to install a new tool into the
       public database.

   ◦   Make New Category - Allows the user to make a new category
       to appear on the RDE menu bar.

   ◦   Trash Can - Only the System Administrator has approval to re-
       move trash. (Figure 27, "Administrative Password Retrieval,"
       on page 42) Tools dragged to and then dropped into the trash
       can will be removed from the database permanently. Once it's
       removed, no other users will have access to it in their RDE ses-
       sions.

Make New Category

Figure 14, "Make New Category," on page 30 allows the user to cre-
ate new category tool palettes. After adding new categories (i.e. Util-
ities, Software, and Telnet) to the RDE, the menubar will change to
contain the following displayed categories (See Figure 15, "Menu
Bar (with newly added categories)," on page 30).
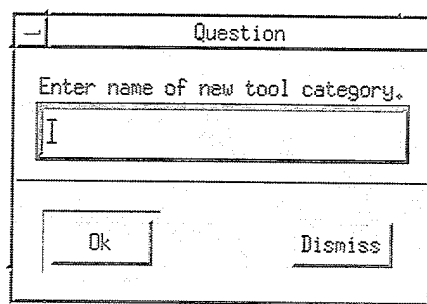
FIGURE 14.   Make New Category

   ◦   OK - Make a new category and add its name to the RDE
       menubar.
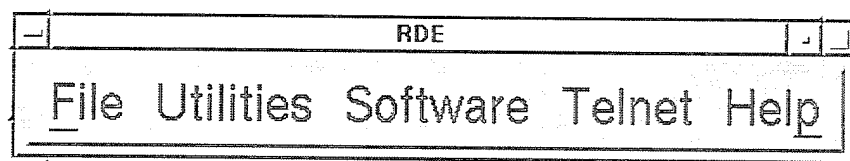
FIGURE 15.   Menu Bar (with newly added categories)

Show Function Names
Show Tool Names

Toggle the labels next to all icons in the Tool Database window, as well as the tool palettes, from either the actual tool name to the generic function name, or the function name to the actual tool name.

Install New Tool

Installing a new tool will normally be done by a super user/SysAdmin type person, but it is possible by anyone in this release. If there is a tool that a user thinks should be made available to all the users of the RDE, and that tool is not already in the Tool Database (it doesn't have an icon in Figure 13, "Show All Tools (Tool Database)," on page 29),the dialog in Figure 16, "Install New Tool Window," on page 31 would be used to enter all of the information necessary to install a new tool into the database.

It is possible to install a new tool by ONLY providing a Tool Name and a Launch Command, since everything else can be defaulted. This is not recommended, however, as the more information that is accurate, the better the tool is represented.

FIGURE 16.   Install New Tool Window

○ **Default host/machine** - The machine that will be used to run
the tool each time that it is launched.

○ **X application?** - Is the application an X Window application?

○ **Add to Database** - Adds this tool to the public database.

○ **Input TES file...** - Prompts the user for a TES file. The TES file
is used to fill in some of the fields of the form.

○ **Help** - Pops up a window with a short description of each field
to be entered.

Figure 17, "New Database Tool Adder (Help)," on page 32 is the
help window that is popped up after hitting *"Help."* It provides short
descriptions of the fields that describe the attributes of a tool being in-
stalled.


```
┌─┐                              Message                                │
│ Tool name:          Name that will appear next to the icon.
│ Function name:      The general function that the tool performs.
│ Launch command:     The binary/executable name.
│ Options/switches:   Any options or switches that are passed as arguments.
│                     If one of the options or switches is the name of a
│                     dynamically specified file, then insert the keyword
│                     FILENAME where the file name would be inserted at run time.
│ Pixmap file:        Name of the pixmap file that resides in the directory
│                     specified with the environment variable $RDE_HOME/etc/icons.
│                     Specify, ONLY, the .xpm file name, not the full path to it.
│ Man page keyword:   Not implemented yet. Can leave blank.
│ Version:            Version number of the tool. This version number will
│                     be used to distinguish amongst multiple copies of the
│                     same tool in the database. This enables you to have an old
│                     version and a new version of a tool to choose from. How-
│                     ever, it still stands that whatever is specified in the
│                     "Launch command" field will be launched.
│ Default host:       The host that this tool will be launched from. If
│                     'DEFAULT' is specified, then the host that the rde was
│                     launched from, will be the host that is used to launch
│                     the tool.
│ X application?:     'Yes' means that it is an X application, (using the X
│                     Windows System). 'No' means that it is a command-line
│                     type of application (requiring no X Window System).
│
│                            ┌──────┐
│                            │  OK  │
│                            └──────┘
```

FIGURE 17.   New Database Tool Adder (Help)

Quit

When you've finished using the RDE, you can quit out of it to end the current session. Be sure to quit out of the RDE before exiting your window system or logging out, as most of the state of your session is saved only by quitting properly.

- Selecting Quit from the File menu pops up a window to confirm your decision. DO NOT use the window manager Destroy or Delete choice from the window manager's menu, or an unfavorable exit will occur, and no changes will be saved to the user's personal configuration file.
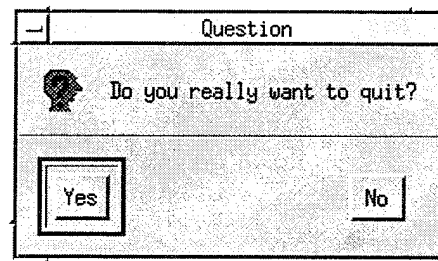
FIGURE 18. Quitting the RDE

Menu Bar Categories

You can now choose commands from the menus in the menu bar. To get the list of tool icons for a category, choose *"Tool Palette"* from beneath a category name (for example, Utilities as shown in Figure 19, "Menu Bar (selecting empty category tool palette)," on page 33).
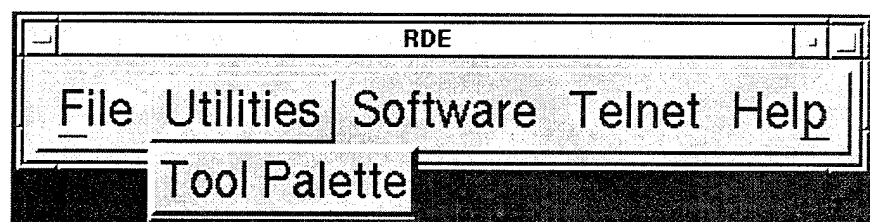
FIGURE 19. Menu Bar (selecting empty category tool palette)

This brings up a blank tool palette - if the category hasn't been filled.
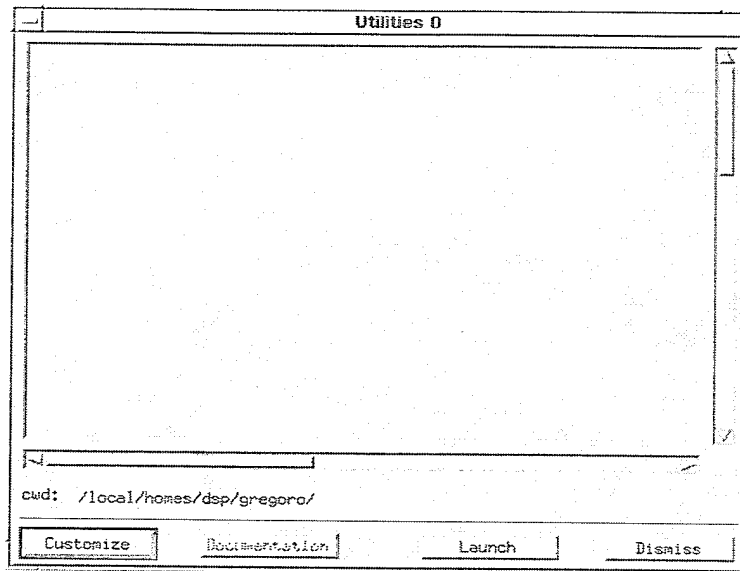
FIGURE 20.   Category Tool Palette (empty)

Using a Tool Palette

When you select a category (by selecting the *"Tool Palette"* choice from the category menu) from the RDE menu bar such as Utilities, Software, Telnet, or some other custom category, a tool palette is displayed. Multiple instances of the same category tool palette can be displayed. The idea behind this is that the user might want to have one Software tool palette in one virtual window and another Software tool palette in another virtual window. This is only useful if your window manager supports virtual windows.

For each tool palette instance, the window is numbered, to be able to distinguish multiple instances from each other at quitting time. These numbers are only needed for helping distinguish the windows when remembering the window layouts.

From the Tool Palette, and from the Tool Database window, you can view information about a tool prior to launching it, and you can directly launch a tool by double-clicking on it. To view information about a tool prior to launching it, see Figure 25, "Tool Configuration (Tool Information)," on page 39. To launch a tool without first viewing its information, see Figure 31, "Tool Palette with Popup Menu," on page 44.

NOTE: The tool palettes for any category have the same command functionality, i.e. they have the same command buttons and subsequent behavior. The only difference is the tool set defined within the tool palette for a category. (The category, Software, might contain debugging aids or compilers for developing software while a category, Project Management, might contain scheduling and cost accounting aids for managing a project.)

Adding a Tool to a Category

With the middle mouse button, drag an icon from the Tool Database window to a desired category Tool Palette. Then, letting go of the button to add that tool to the receiving category. See Figure 21, "Adding tools to the new tool palette (dragging with Button2)," on page 36.

Removing a Tool from a Category

With the middle mouse button, drag an icon from a category Tool Palette to the Tool Database window. Then, letting go of the button will remove that tool from all instances of the category that contained it, and put it back into the tool database palette. The icon in the Tool Database window will now be un-grayed-out, since while it existed in a category tool palette, it was grayed-out in the Tool Database window.

Changing a Tools Category

With the middle mouse button drag an icon from one category to another. When you let go, the tool will be removed from all the category tool palette instances that used to contain it, and will appear in all the tool palettes of its new category.
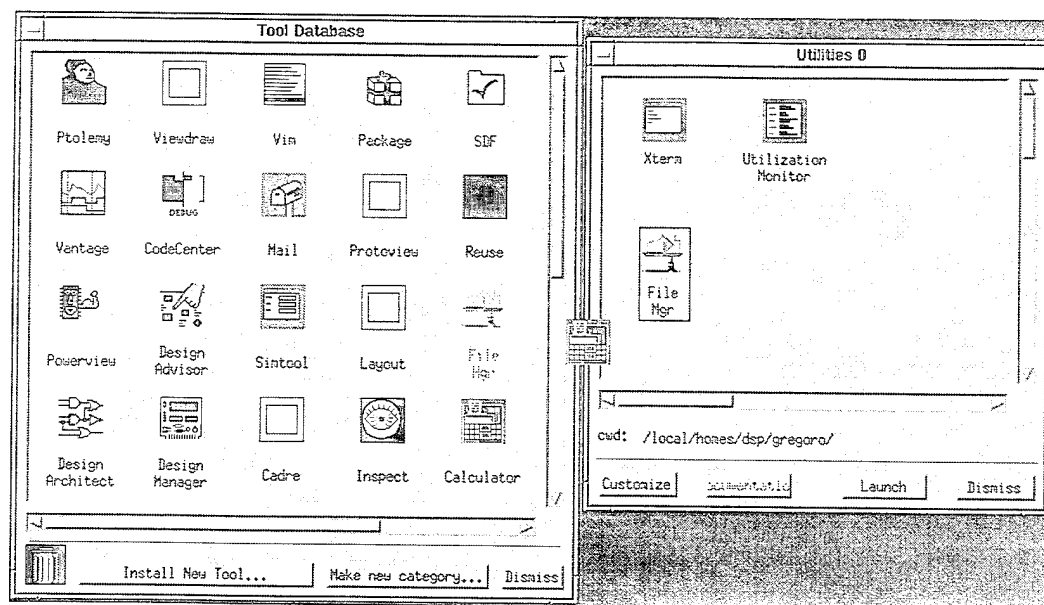
FIGURE 21.   Adding tools to the new tool palette (dragging with Button2)

The tool palette will then have some tools in it:
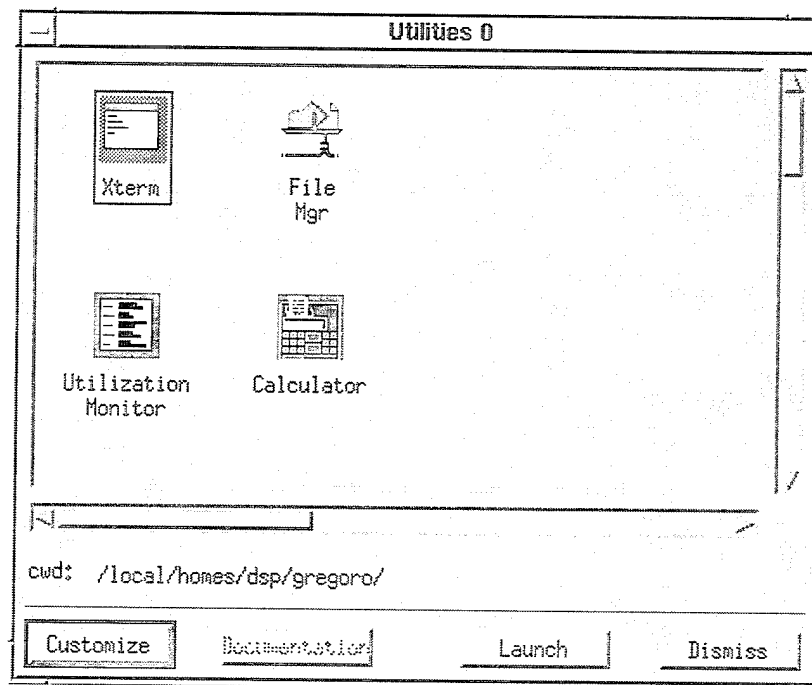
FIGURE 22.   Newly filled Tool Palette

And the menubar choice, Utilities, will now have these tools under-
neath it - which will be available for launching by selecting them:
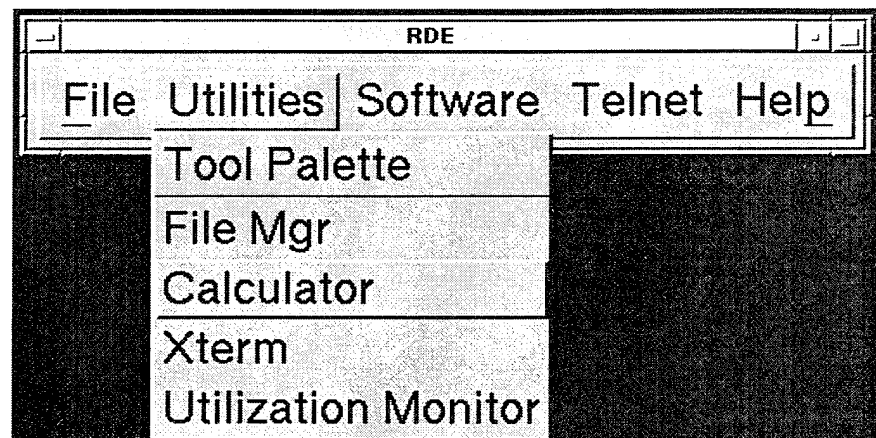
FIGURE 23. Tool choices corresponding to icons in new tool palette

To select a command from the menu bar, single click the menu name,
and then single click the menu choice. Or select the menu name, and
while holding Button 1, drag down to the menu choice and release the
button.
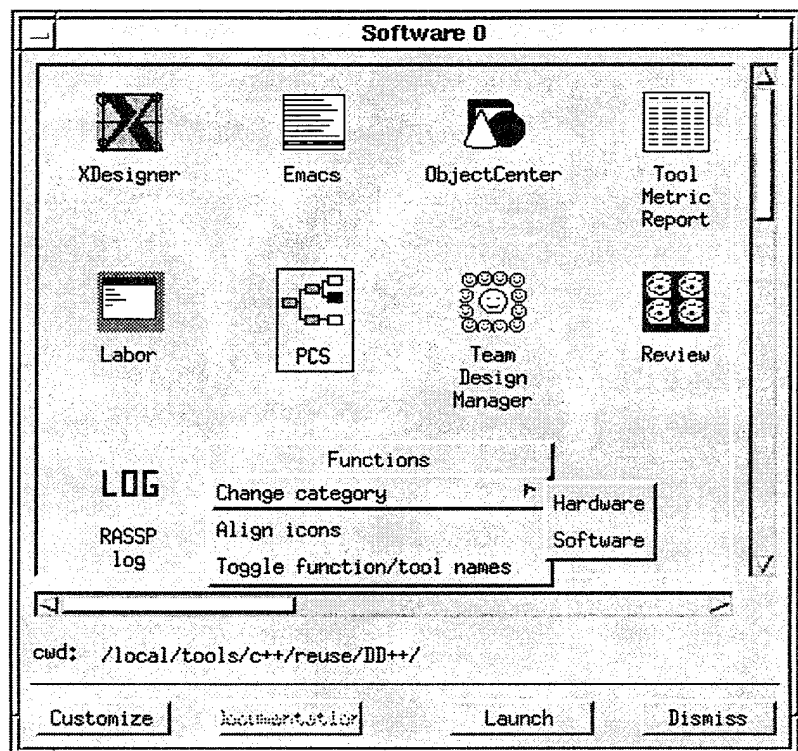
FIGURE 24. Tool Palette

- ° **Customize** - View and possibly customize the attributes of the current tool.

- ° **Documentation** - Not yet available, this option will show the documentation for this tool.

- ° **Launch** - Run the selected tool on the default host.

- ° **Popup** - Button3 - When pressed on the background of the palette (not on a tool) the pop up Functions menu appears.

- ° **Change Category** - When dragging over the 'Change category' menu choice, there should be a sub-menu that pops up. It will contain the list of categories that already exist for your rde session. Choosing one of these will swap out the current category of tools, with the newly selected category of tools.

- ° **Align icons** - Move each icon to its own spot on the invisible grid.

- ° **Toggle function/tool names** - Replace the names underneath the icons with the generic descriptive name of the appropriate tool, if the specific tool name is currently being displayed, and vice versa.

Customizing a Tool

The Tool Information window is selected by choosing the Customize option in the popup menu or the Customize button as shown on Figure 24, "Tool Palette," on page 37. When you select a tool in the tool palette, the tool gets a square border around it to indicate that it is the *"active"* tool.

This window displays information about the tool configuration, see Figure 25, "Tool Configuration (Tool Information)," on page 39, including its version, name, function, and icon pixmap file. You can modify the default information and save it as the new default. Personal saving occurs in the user's own configuration file, it does not affect the attributes of the tool in the database - so making changes will not bother other users. A public save will eventually affect other users when they start up again - and a password is required to be able to do a public save.

Tool icon —

Current file box —

Icon pixmap file —
box

Launch Command —
box
Switches/Options —
box

Host button —

Version button —

```
┌─────────────────────────────────────────────────────────────┐
│ ─┘              Tool Information                             │
├─────────────────────────────────────────────────────────────┤
│  LOG                    RASSP log                      LOG   │
│                                                             │
│  Name                           Function                    │
│  │RASSP log                     │Metrics                    │
│                                                             │
│  Menu accelerator                                    Char   │
│  ☐ Meta  ☐ Ctrl  ☐ Shift  ☐ Alt  ☐ None  │        │       │
│  Current file                                               │
│  │                                                          │
│                                                             │
│  Icon pixmap file                                           │
│  │rassplog.xpm                                              │
│                                                             │
│  Launch Command                 Switches/Options            │
│  │rassplogtest                  │$FILENAME                  │
│  rassplogtest FILENAME                                      │
│                                                             │
│      Host     dopey    ─┘          Version  1.0 ─┘          │
│                                                             │
│  ┌──────────────────┐  ┌─────────────┐  ┌───────────────┐   │
│  │Revert To Public Info│ │ Public Save │  │ Personal Save │   │
│  └──────────────────┘  └─────────────┘  └───────────────┘   │
│  ┌──────────────────┐  ┌─────────────┐  ┌───────────────┐   │
│  │ Output TES file...│  │   Launch    │  │   Dismiss     │   │
│  └──────────────────┘  └─────────────┘  └───────────────┘   │
└─────────────────────────────────────────────────────────────┘
```

FIGURE 25.   Tool Configuration (Tool Information)

- **Tool Icon** - The tool icon is customizable. To modify the icon, you need to either update the actual file that is displayed in the 'Icon pixmap file' box, or replace the pixmap file name with that of another pixmap file name. The 'Icon pixmap file' box contains only the name for the pixmap file as it appears in the $RDE_HOME/etc/icons directory. This directory contains pixmaps for all tools in the database, so when a new pixmap is desired, it must be put into this directory to be accessible.

°   **Version Button** - The Version button displays the version of
    the tool. Eventually (in later releases) if other versions exist,
    they will be displayed in the pop-up menu when you click on the
    version button - but for now, multiple versions are represented
    by multiple icons. Multiple versions are useful when you have
    different files using different versions of the same tool.

°   **Current File** - The Current file box will display the currently
    selected file. This file will be used as a switch/option when the
    tool is launched. It replaces $FILENAME.

°   **Launch Command** - The Launch Command box contains the
    command used when the RDE launches the tool. It uses this
    command in conjunction with the options contained in the
    Switches/Options box. For example, to launch a file in Object-
    Center, you may always want the options to be objectcenter (for
    the launch command) and, **-l** $FILENAME **-g** (for the Switches/
    Options text). $FILENAME will be the value of the entry in the
    "Current File" text box.

°   **Host Button** - The Host button contains an option list of hosts
    denoting from where you can launch the tool.

°   **Public Save** - Save the changes, just made in this window, to
    the database. The database is a public database, so the changes
    made to it will be available to all other users when they run the
    rde. An administrative password is required in order to make a
    public save, so if it is needed, then it can be found out by the rde
    administrator.

°   **Personal Save** - Save the changes, just made in this window, to
    the user's personal configuration file (in his/her home directo-
    ry) named ".rdetools". These changes won't be available to oth-
    er users, and so the user can experiment without fear of
    affecting other rde users. If the user desires to reset the tool in-
    formation to what is in the public database, then the Revert To
    Public Info button can be used.

°   **Revert To Public Info** - If the user desires to reset the tool in-
    formation to what is in the public database, then this button will
    wipe away the attributes currently associated with the tool, and
    replace them with the ones currently being held in the public da-
    tabase.

- **Output TES File...** - This will prompt the user to specify a directory and file name in which to output the TES file. The output of a TES file is a step in an effort to have the rde completely compatible with the industry standard of tool encapsulation techniques. Outputting a TES file, as the rde is currently implemented will produce a minimal, but accurate, TES file that can be read by other TES parsers/readers. The rde will currently read in a TES file, if the user specifies so, when a new tool is installed. A more robust and accurate TES reader is scheduled to be implemented by a member of CFI (CAD Framework Initiative), and will eventually be integrated with the rde to provide a more useful interface to TES files. This is not a necessary function for the user to use - it is strictly optional.

- **Launch** - Launch the tool.

  NOTE: If the host, from which you wish to launch tools, is missing from the hosts lists, contact your system administrator.

Saving Tool Information

If you made any changes to the tool configuration in the Tool Information box that you wish to be the new default, you should save your changes. The changes you make can either be reflected in only your environment, or in the public environment. If you wish to make a change to the public environment, you will be prompted to enter a password first.

Note: Only the System Administrator has approval to change.

Personal Save

Save the changes made to the tool to the user's rdetool file in his/her home directory. The following window appears when the change is made:
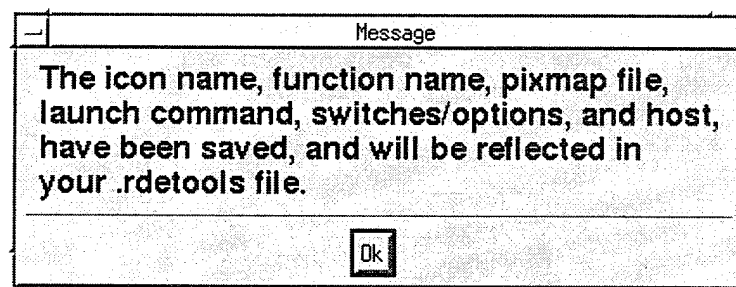
FIGURE 26. **Message Window - Personal Save**

This message states that changes made to the Tool name, Function, Icon pixmap file, Host selection default, Launch Command and Switches/Options text, will be saved to your .rdetools file.

NOTE: This message is displayed momentarily. It will be dismissed automatically unless you click OK to dismiss it manually.

Public Save

If the administrative password has not already been given, then the following prompt will be popped up to let the user enter it:
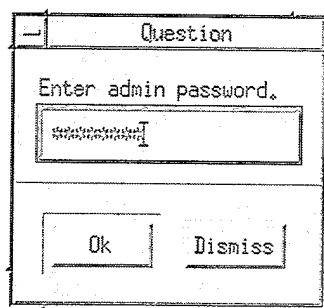
FIGURE 27.   Administrative Password Retrieval

The rde administrator will type the password. If the correct password is entered, the following popup is shown:
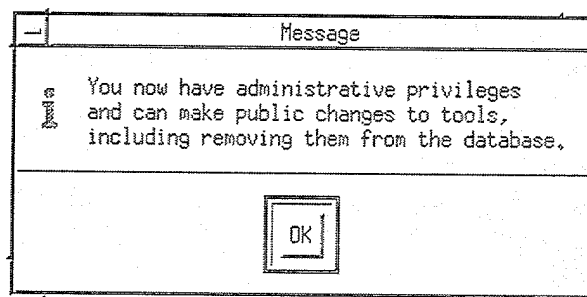
FIGURE 28.   Administrative Password Retrieval - OK

Then a loop will be entered, allowing the user to re-enter the password until it is correct, or until the user gives up. The following popup is shown when the wrong password is given:
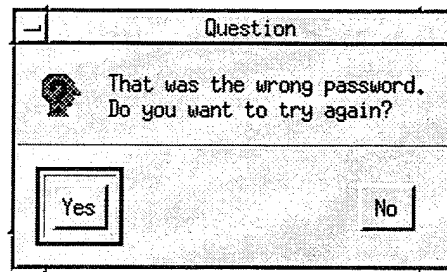
FIGURE 29.   Administrative Password Retrieval - WRONG

Finally, when the correct password has been given, the user may save the changes made to the tool, to the database, where the changed tool will be available to every user who starts up the rde in the future. The following window appears when the change is made:
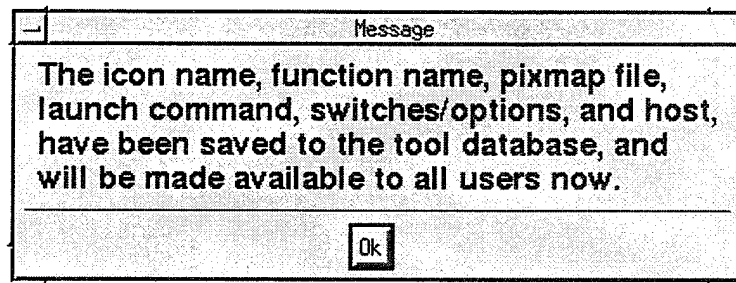
FIGURE 30.   Message Window - Public Save

This message states that changes made to the Tool name, Function, Icon pixmap file, Host selection default, Launch Command and Switches/Options text, will be saved to the database, which is read by everyone at start-up time.

> *NOTE: This message is displayed momentarily. It will be dismissed automatically unless you click OK to dismiss it immediately.*

Launching Tools

You can launch a tool from several windows of the RDE: from the top menu category pull-downs, any tool palette, the Tool Database window, and from the Tool Information window. To launch a tool from a Tool Palette or from the Tool Information window, see "Launching a Tool from a Tool Palette" and "Launching a Tool from the Tool Information Window" sections which follow.

Launching a Tool from a Tool Palette

Using Button1, double-click on the tool icon to launch it from the default host. Or, using Button3, popup a Functions menu, from which the host is selected from the Launch submenu.
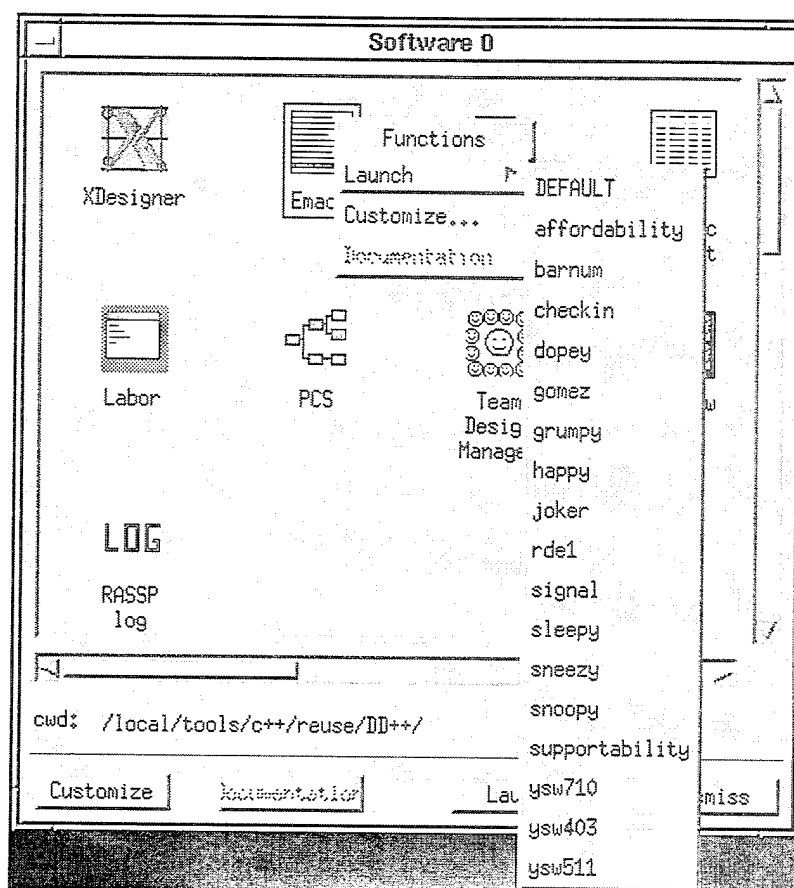
FIGURE 31.   Tool Palette with Popup Menu

- **Popup** - Button3 - When clicked on a tool, this Functions popup menu appears. When dragging over to the right of the launch choice - this host list appears. Selecting a host will launch the tool off of that host.

- **Launch Button** - Click on the launch button at the bottom of the window, and the "active" tool (one with box around it) will launch from the DEFAULT host, which is the host on which the rde was started, or a specific host that was set as the default for that particular tool.

Launching a Tool from the Tool Information Window

The Tool Information window is displayed after selecting "*Customize*" from either the bottom of the tool palette or the Functions popup menu of the tool icon. Hit the "*Launch*" button to launch the tool. The tool will launch using the information shown - launch command, switches/options, and current file (if $FILENAME is an option).

NOTE: For information on how to use the launched tool, follow the user's guide that accompanied the tool.

Help Menu

The RDE Help command provides:

• Information about the RDE

• Help Desk capabilities

• Screen sharing capabilities

• Problem Reporting capabilities
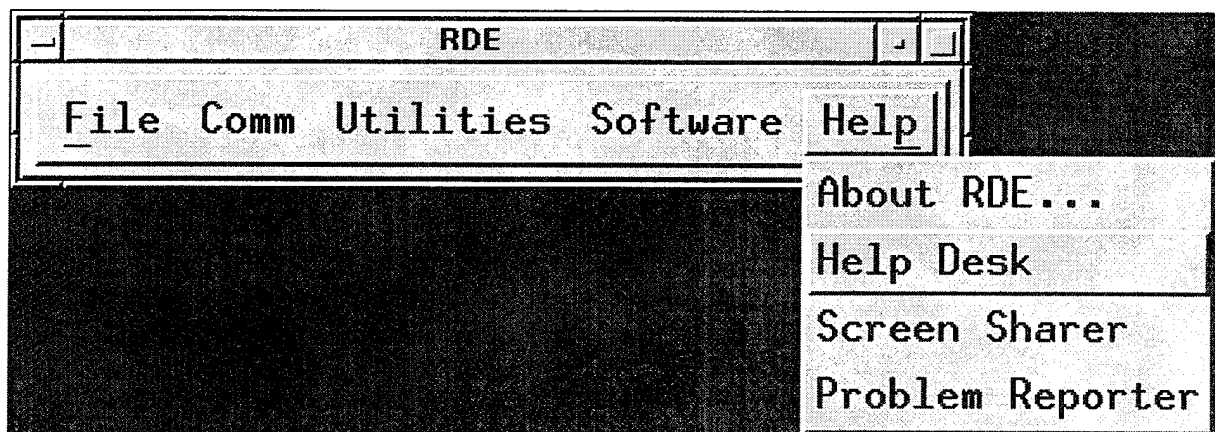
FIGURE 32.   **Using Help**

Using About RDE

Within the Help menu, you can view information about the RDE, e.g., version and release date.
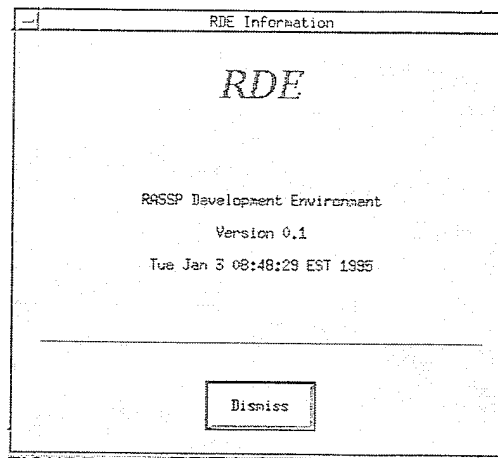
FIGURE 33.   RDE Information Window

Using the Help Desk

Invoking the Help Desk will provide on-line help. Currently this help is provided via the WWW, and invoking the Help Desk will open up a web browser to the site containing the documentation available. If there is no web browser available, the only documentation is this Users Guide. Additional help may be obtained through customer support (see the section "Customer Support" in Chapter 1).

Using the Screen Sharer

Invoking the Screen Sharer will try to invoke CECED if it is available on your system. It will invoke it with a session of the rde running on it in order to use the rde in a screen-sharing capacity.

Using the Problem Reporter

Invoking the Problem Reporter will invoke the Problem Report utility that is discussed in the Problem Report Tool chapter.

Chapter 3: The Product Structure Editor

Chapter Content

Refer to this Chapter for:

Introduction
System
View
Relationships

Introduction

Note to Reader: The PSE, including this version, is currently being debugged and continuously developed. Do not perform a formal test on this, due to the debugging being done at Sanders, but we would appreciate any bug reports or new features that you would like to see. Please contact us with any problems at robc@sanders.com. Thank you.

GENERAL NOTES

- Use the TAB key or point and click with the mouse to move between the fields of a window.

- Selecting the Dismiss button displayed at the bottom of windows will close the current window.

Starting the PSE

Type **pse** in an xterm window and press Return.

The main gui is the first interface which appears when PSE(Product Structure Editor) is run. There is three main options: System, Views, and Relationship Figure 4, "Information View," on page 50.
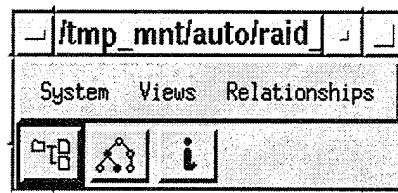
FIGURE 1.   Main Gui of the PSE

System

Under Systems, there are two options: Translator Setup and Exit. Translator Setup allows the user to define a format translator Figure 4, "Information View," on page 50. Currently, a list of translators is read in from a file; however, they will soon be stored in the database. Right now, this setup gui will add the information to the file and refill the tables, by rereading the file. Later this setup gui will just add the information to the database. The first field expects the command name of the translator. The

second field expects the file extension that would exist on the file to be translated: ex. gif, xpm. The third field expects the new extension of the translated file: ex. tiff, ppm. The last field would contain any flags needed to run the translator: ex -o. Exit would, of course, exit the PSE.
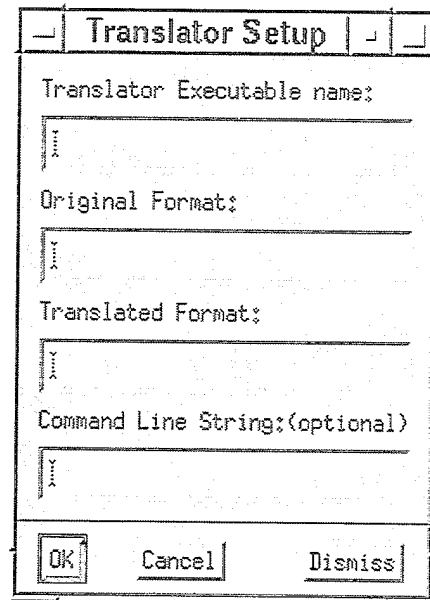
FIGURE 2.   Translator Setup

Views

Under Views, there are three working options: Parent/Child(Hierarchy), Information, and Browse. The first pops up a Parent/Child, or hierarchical, view Figure 4, "Information View," on page 50. Basically, this shows a center icon and its parents and children. This view type may also be created by pressing the icon with hierarchy chart on it(second one). Information view shows textual information about an object, usually the center object Figure 4, "Information View," on page 50. This view may also be created by pressing the `i' icon. Another view is currently in the works and it is a browsing view Figure 4, "Information View," on page 50. To traverse this view siply single click the icons, not double click. (More on this as it develops) More on the view interface is provided below.

View Dialog

View   Relation   Bookmarks

pseMainGui_xd.h

pseRelationSetting_xd.h

Folder1

    driver.cc

    Makefile_g++

Makefile_g++

FIGURE 3.   Browser View

```
┌─┐                          View Dialog                    ┌┴┐ ┌─┐
│─│                                                         └─┘ └─┘
├───────────────────────────────────────────────────────────────┤
│   View   Relation   Bookmarks                                  │
│  ┌──────────────┐                                              │
│  │ ⊘ │ ⊞⊟ │                                                    │
│  └──────────────┘                                              │
│ ┌────────────────────────────────────────────────────────────┐│
│ │                                                            ││
│ │   Name: Folder1                                            ││
│ │   File:                                                    ││
│ │   Format: rle                                              ││
│ │   Subscriptions: prime                                     ││
│ │   creation_date: 10/06/95                                  ││
│ │   creator: prime                                           ││
│ │   0 -> driver.cc                                           ││
│ │   0 -> Makefile_g++                                        ││
│ │   0 <- ROOT                                                ││
│ │                                                            ││
│ │                                                            ││
│ │                                                            ││
│ │                                                            ││
│ │                                                            ││
│ └────────────────────────────────────────────────────────────┘│
└───────────────────────────────────────────────────────────────┘
```
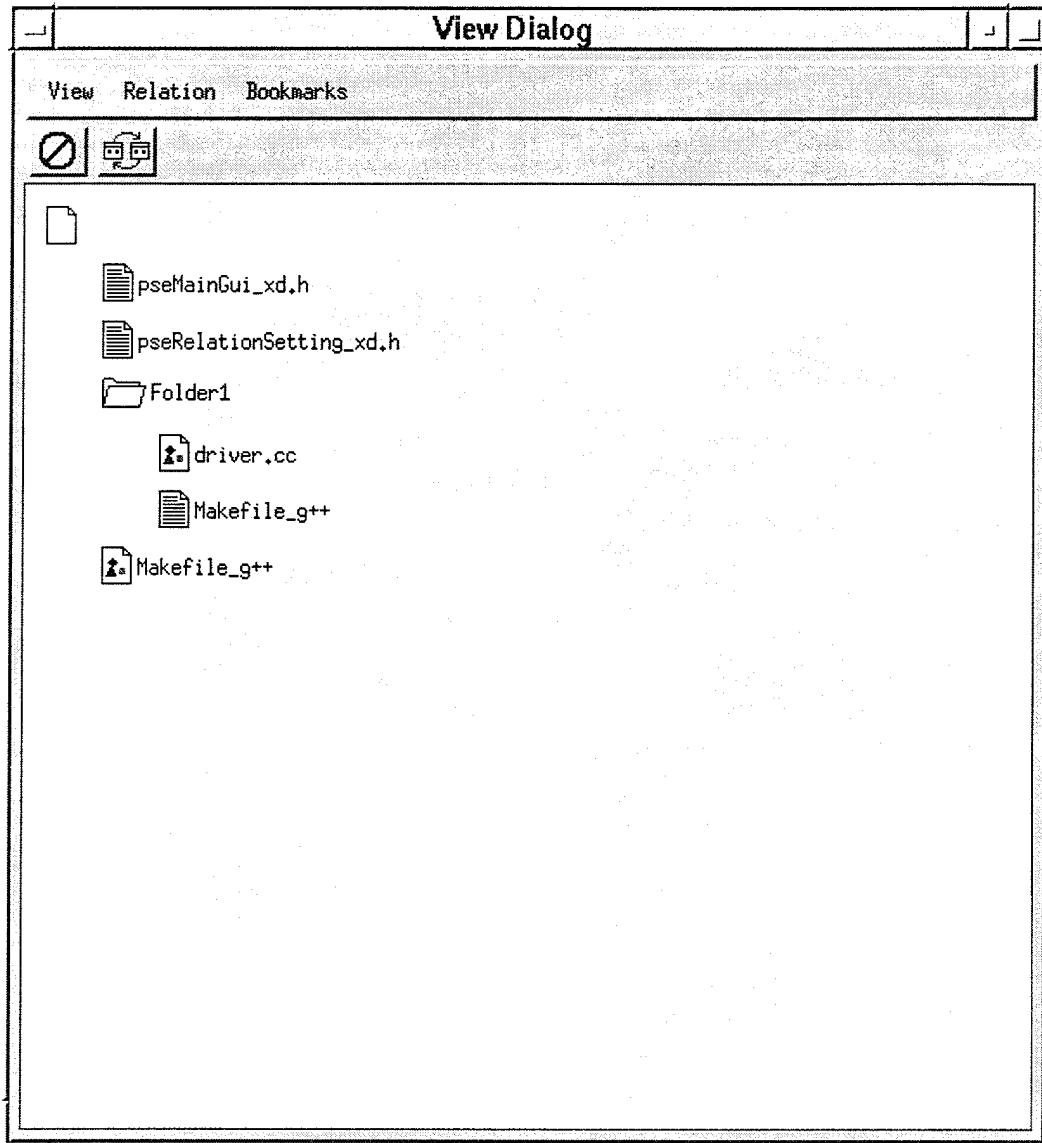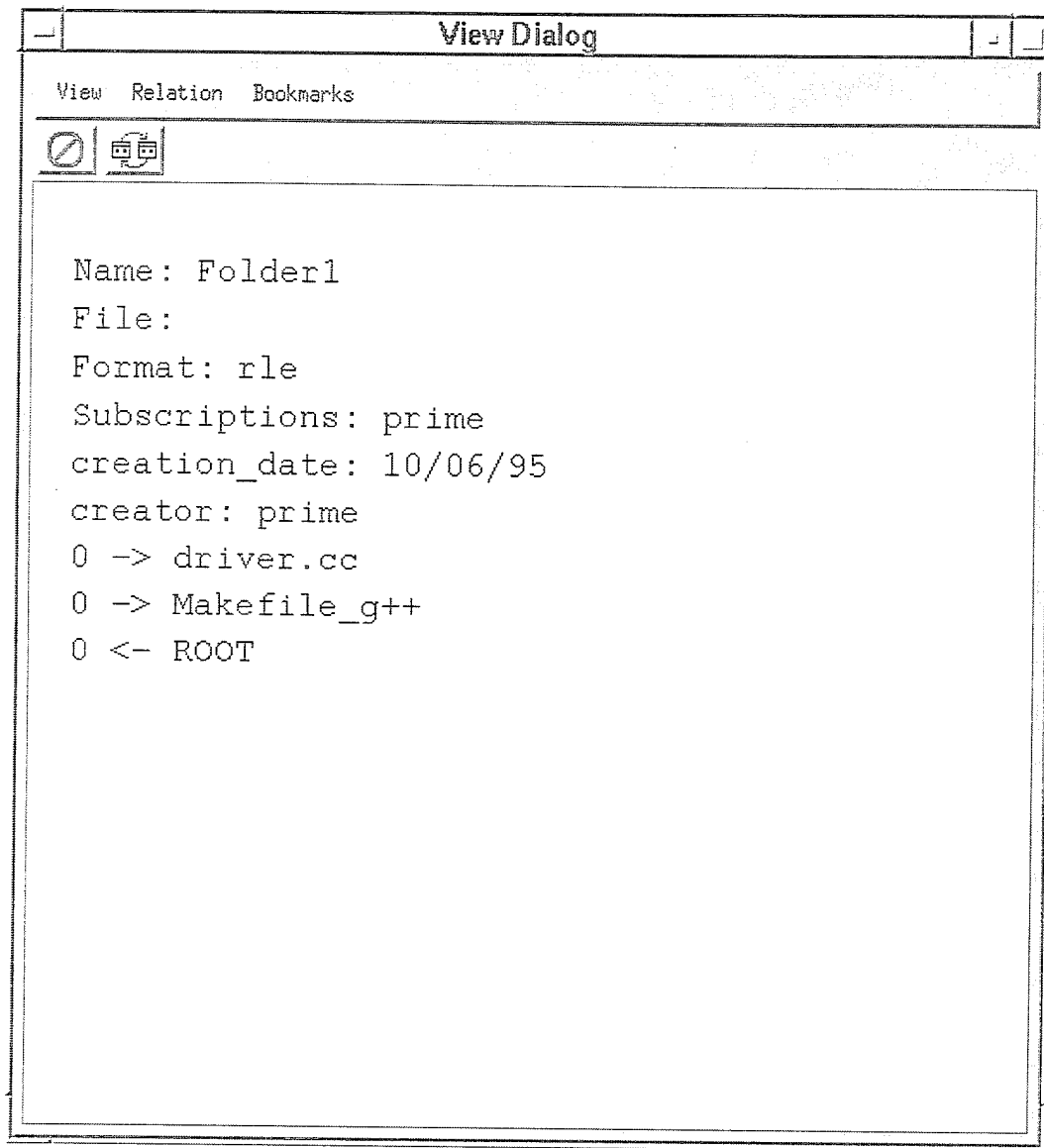
FIGURE 4.   Information View

Relationships

Under Relationships, there are three options:Edit Types, Set Relation, and Edit Objects. Edit Types pops up a gui within which new relationship types may be added, edited, and in the future deleted Figure 4, "Information View," on page 50. Relationship types should be defined before any are set. Set Relation pops up a gui within which two objects may have a relationship defined about them Figure 4, "Information View," on page 50. Using the second mouse button, one object icon is dragged onto

the `From' drop site. Then, another object is dragged onto the `To' drop site. The dragged icon should go from red to green, green indicates a legal drop site. The relationship is selected from the list given. Edit Objects pops up a gui which allows new object types to be added to the database Figure 4, "Information View," on page 50. When PSE is first installed, there either should be types added through this gui or imported through a command line import. The common object types are folder, document, image, and workflow.
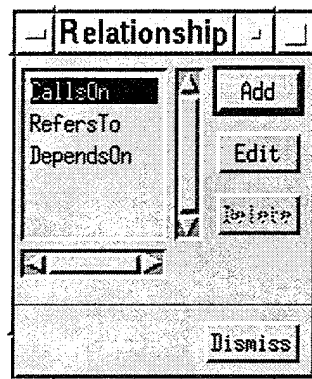
FIGURE 5.   Edit Types Gui
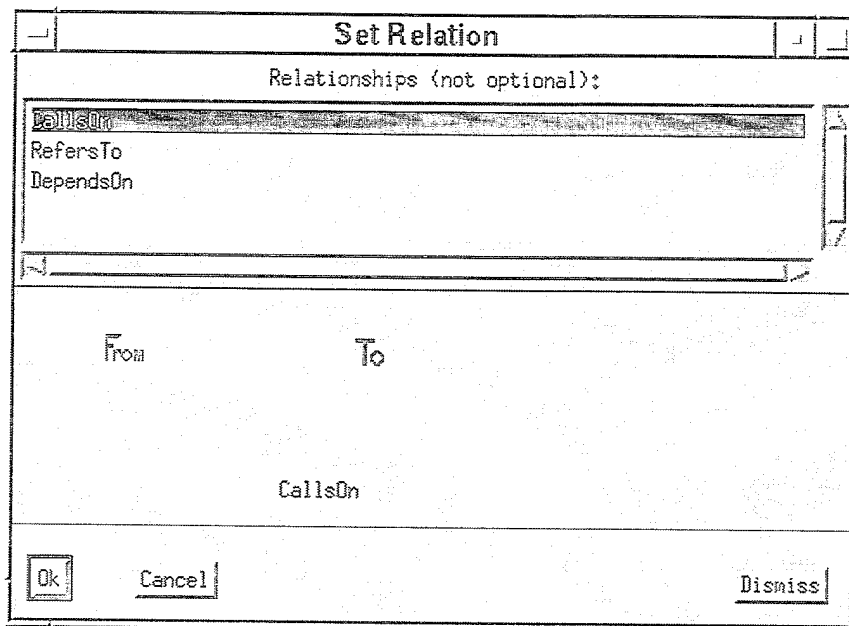
FIGURE 6.   Set Relation Gui
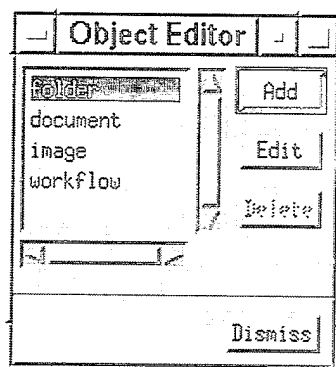
FIGURE 7.   Edit Objects Gui
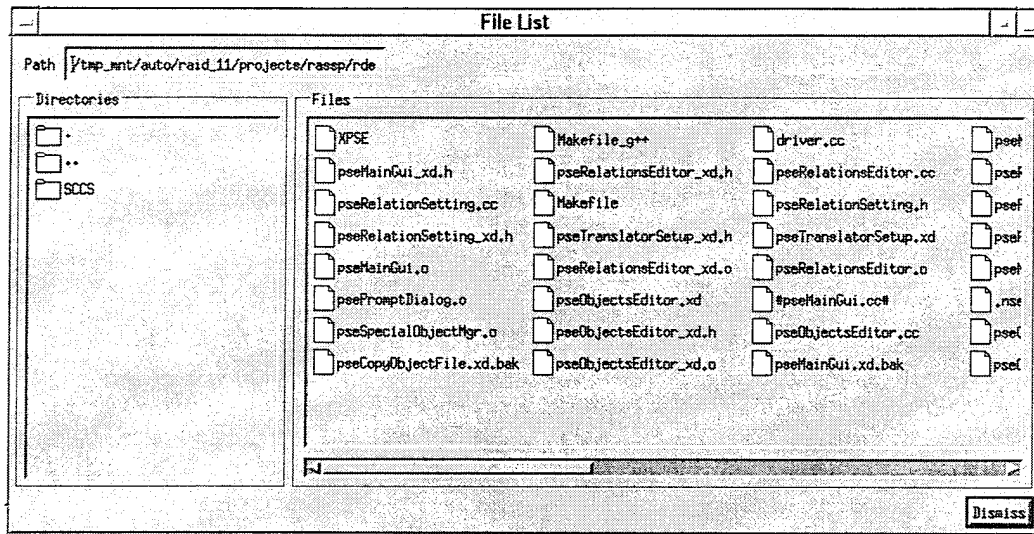
FIGURE 8.   Directory Browser Gui for grabbing files

The first icon on the tool bar is used to popup a file browser type dialog Figure 4, "Information View," on page 50. This is used to get files from directories, by dragging them onto the icons in a view, using second mouse button. The second icon brings up the Parent/Child view. The third icon with the `i' on it brings up the Information view.

On the views, there is another menu bar with three options: View, Relation, and Bookmarks.

Under View, there is a list of all the possible views that the current one can be converted to. Relation currently contains `Hierarchy' and `Depends On'. Pressing one would adjust the order of the icons on the view, if it applied to them. In the future, this should be updated to contain all possible relationships. Bookmarks allows bookmarks, or place keepers, to be added or there label edited. Existing bookmarks for a user should also show up as menu buttons as they are added. Currently, the edit feature is not implemented. Pressing on a bookmark button should force the view to goto that center icon and show the view as it was seen when the book mark is set. The user sets a bookmark by going to the center icon view that it wants to remember, and then pressing the Add Bookmark button.

Two icons on the views are for dismissing the view, red circle with line,

and creating a link to another view, which must be popped up first. When linking views, press the link icon, second one, and the cursor will turn into a link symbol. Next, press the link icon on the other view. This will cause the views to be framed by a matching color, indicated that they are linked. Changing the center object on one, will automatically change it on the other. To choose a center object on the Parent/Child view, simply double click on the icon.
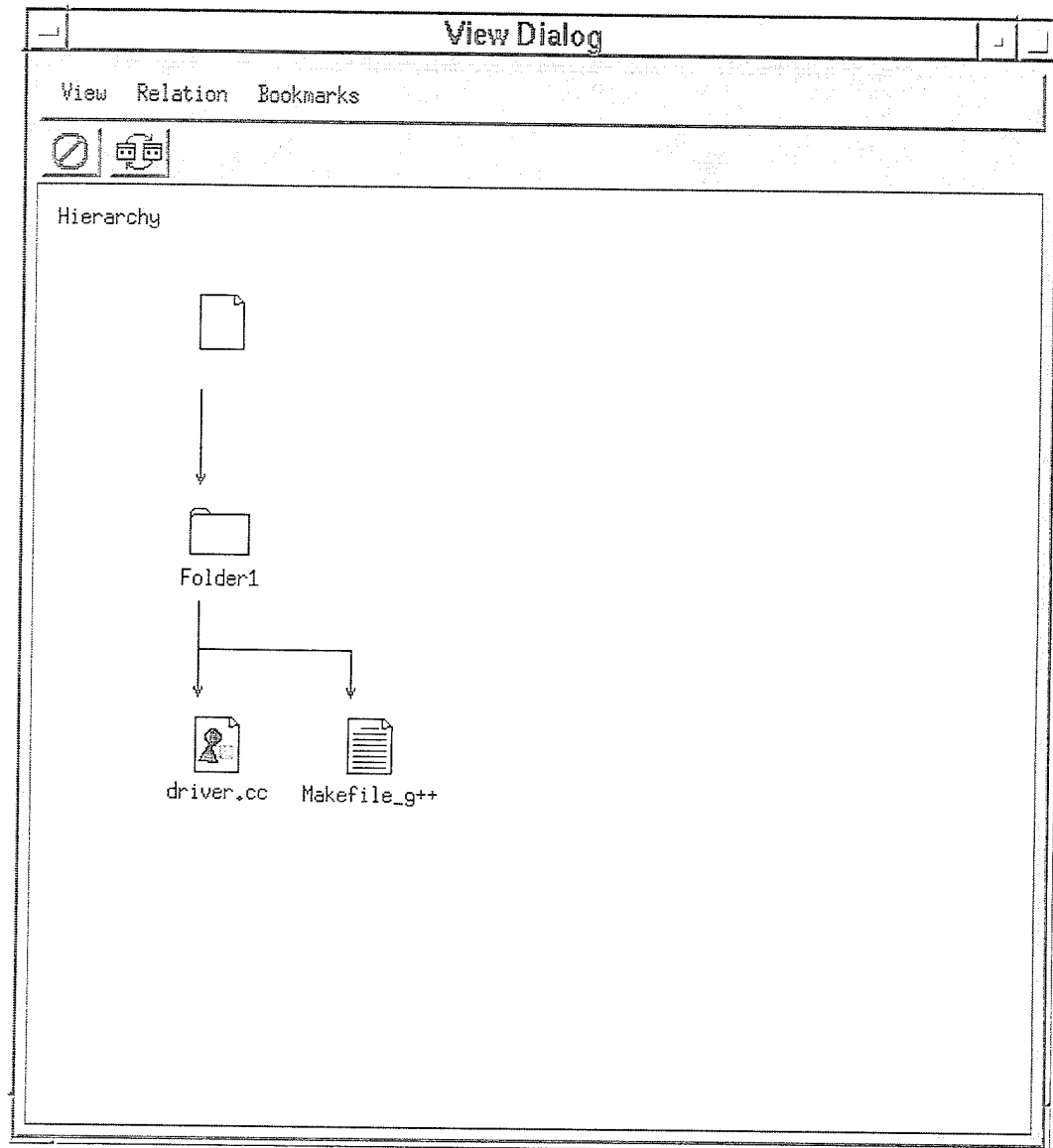
FIGURE 9.   Parent/Child View

Also available on the views, a popup menu appears when the third mouse button is pressed on a icon on the view(not one of the tool bar icons). This menu includes actions for that icon. These may include Check In, Check Out, Subscribe, Edit Attributes, Delete, and Create Folder. Check In will check the file into the database(might not be working yet). Check Out will check the file out of the database(might not be working yet). Subscribe allows the user to add his name to a list of people who will be mailed messages when this file is checked in. Edit Attributes allows the user to edit attributes. Delete will remove the object. Create Folder will create a new folder off this icon. Additional options may show up if there are other actions which can be done to it. For example, a document can be viewed, edited and printed if the tools are available. In that case, each action would show up on the menu with submenus containing the tool used. Selecting on of those tools should cause the action to happen. This is where the translators come into play. Sometimes a file has to be converted in order to be used by the available tools. Usually image files are involved with translators.

Chapter 4: The Remote Data Access Utility

Chapter Content

Refer to this Chapter for:

- Introduction
- Service Broker / Server
- The RDA Client

Introduction

The RDE Remote Data Access Utility (RDA) is client/server software for accessing the RASSP database. This version consists of a Service Broker, a Server, and a Client.

The Service Broker maintains the list of available Servers. The Server registers itself with the Service Broker and waits for a Client. The Client asks the Service Broker for a Server and receives the necessary information to connect to one. The Server then disconnects itself from the Service Broker and spawns a new Server. This new Server registers itself and is available for the next Client. Once the Client and Server are connected the user logs in, performs database operations, and quits. When the user quits both the Client and Server are ended.

NOTE: There is a Server running in the background so that authorized Clients may connect as desired.

The Service Broker (vdasserv) and the Server (rdaserver) must be running on a Sun before a client (rda) is started. The available database operations are Up, Down, Home, Copy Out, Checkout, Cancel Checkout, Checkin and Impound. Currently, these are implemented using the Team Design Manager (TDM) tool by Cadence.

Service Broker / Server
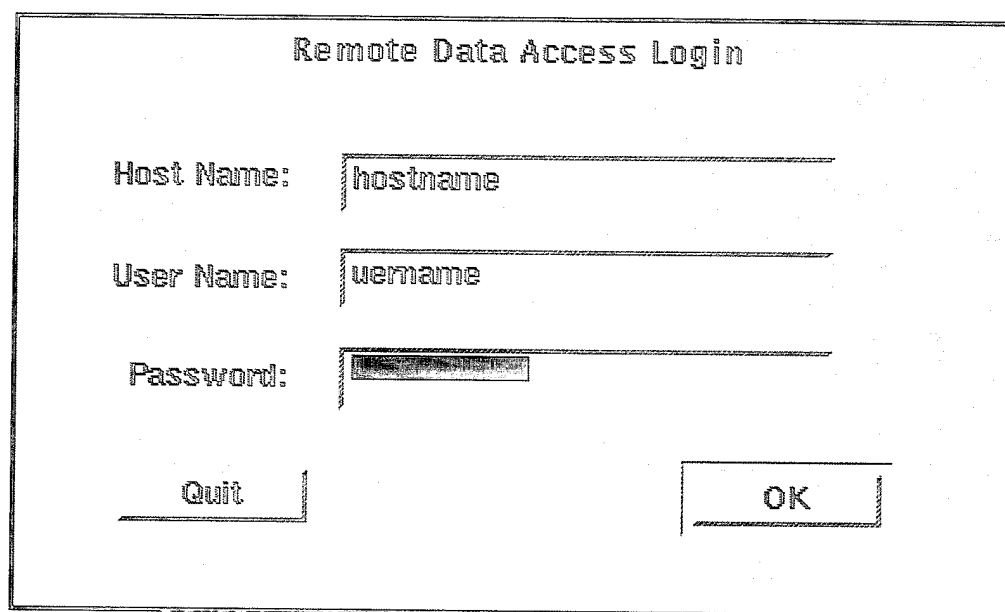
The Service Broker and the Server must be started by your system administrator before the Client is started (and the user logs on).

The RDA Client

The Client software must be started from a directory (folder on the Mac) which contains the executable (rda) and the resource file (rda.vr). On a workstation type: rda -laf motif or launch it from the RDE desktop as you would any other application. On a Mac or PC double-click on the rda icon. The Login Screen will be displayed (see Figure 1, "Remote Data Access Login," on page 58). The user should enter the hostname (fully qualified if it is on a Wide Area Network), login name and password, then press the return key or click on the OK button. If your connection is refused or you get and invalid user message contact your system administrator.

The Main Dialog Screen

Upon successful login the Remote Data Access main dialog window will be displayed, see Figure 2, "Remote Data Access Main Dialog," on page 59. The main dialog screen contains the following areas: the pull-down menu bar; the project name; the remote host name; the current database object; the list of project files; the directory traversal buttons; the database action buttons; the session history; and the Quit button. The pull-down menu bar contains the File menu, which has RDA Info and Quit, and the Projects menu, which has Select Project. Choosing RDA Info will display information about the current version of the RDA client. The other menu entries will be discussed below.

FIGURE 1.   Remote Data Access Login

Quitting

There are two ways to exit the client program. 1) Click on the Quit button at the bottom. 2) Click on the Quit button in the File pull-down menu. With both methods, the user will be asked to confirm the action. The user should click on the Yes button to exit or on the No button to continue the program.

Choosing a Project

The user must select a project before any database actions can be taken. This is done by choosing the Project pull-down menu Select Project option. A list of available projects will be displayed. If the list is empty or incorrect contact your system administrator. When the user chooses a project its name will be displayed on the Project line and a list of all the files in its top-level directory will be displayed in the Project Files window. Directory names will end with a slash (/).
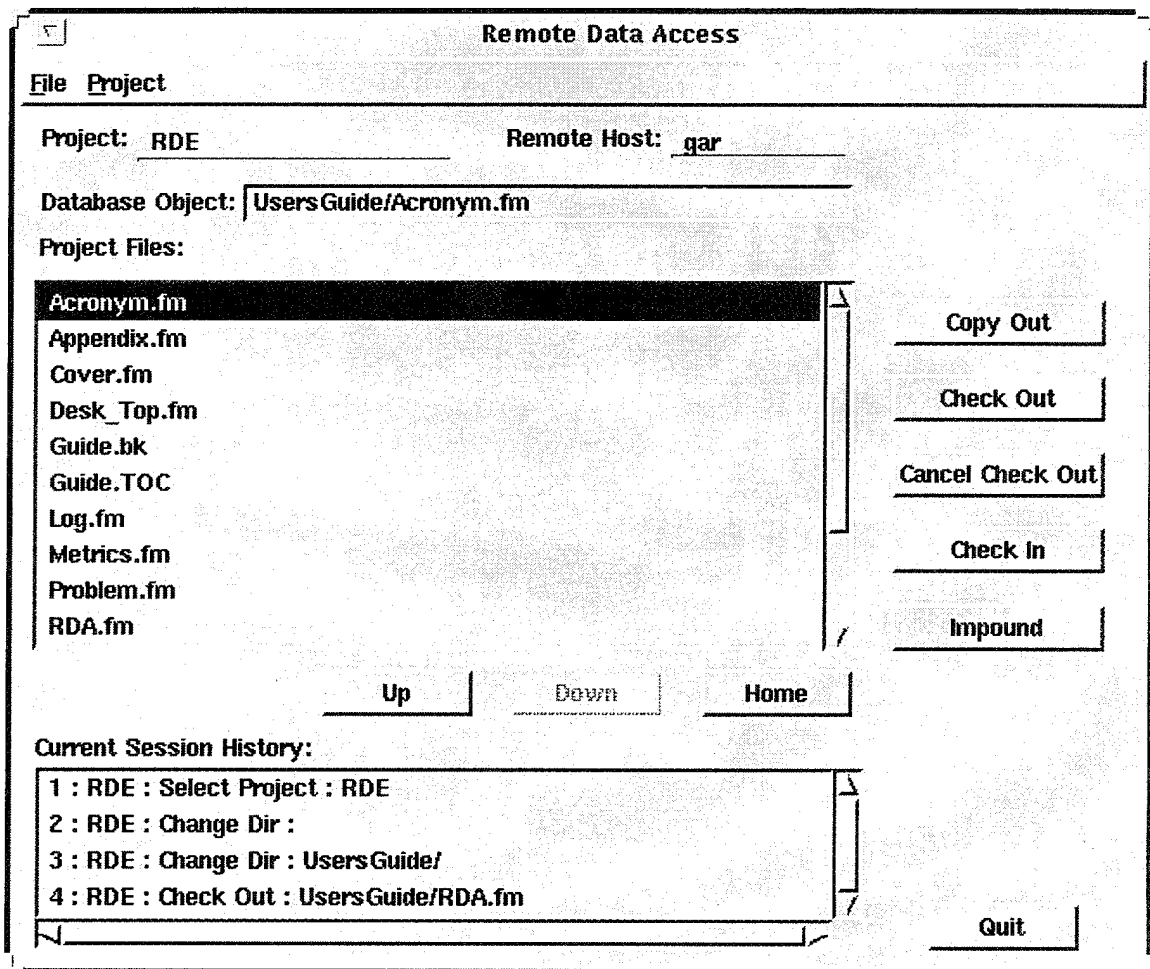
FIGURE 2.   **Remote Data Access Main Dialog**

Choosing a File

To go to a lower-level directory the user clicks on the directory name
and then clicks on the Down button. The user can double-click on the
directory name instead. To go up to the next higher directory level the
user clicks on the Up button. Clicking on the Home button will return
to the project's top-level directory. To choose a file, the user clicks
on the file name. The name will be displayed in the Database Object
window. The user can scroll through a file list by sliding the scrollbar
up and down.

The Copy Out Function

This function makes a read-only copy of a database file to the user's
work area. The user selects a file, then clicks on the Copy Out button,
and the Copy Out window will be displayed, see Figure 3, "Copy Out
Window," on page 4. This window allows the user to browse through
his directories (folders on the Mac) and choose where the file will be
written. The user can scroll up and down using the scrollbar. The di-
rectory levels can be traversed by clicking on the less than (<) and
greater than (>) symbols or by double-clicking on a folder symbol (in
this example there is a folder labeled sub3. The name the file will be
given is displayed in the Save as field.
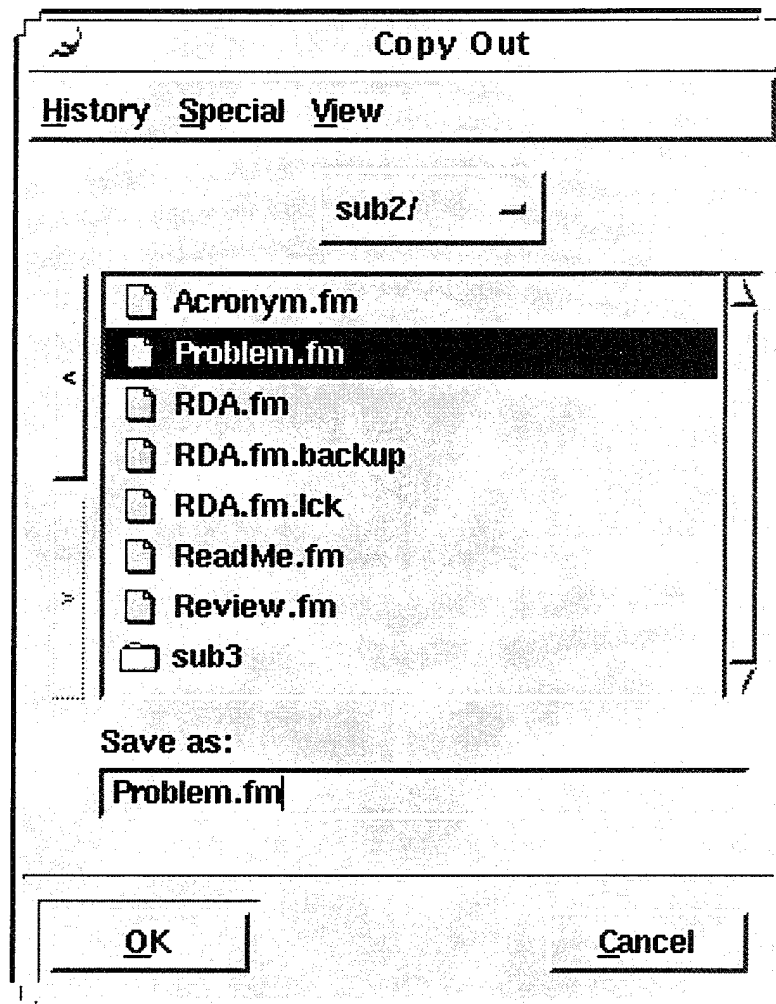
FIGURE 3.   Copy Out Window

After choosing the directory and filename the user clicks on the OK button to start the file transfer, or on the Cancel button to return to the Main Dialog window. While the copy is in progress an hourglass will be displayed and the window will be frozen. Note that if there is an existing file of the same name as the one to be copied out, the user will be notified and must verify the action.

The Check Out Function

This function makes a read/write copy of a database file to the user's work area and locks the file so no other user can check it out. Other users may, however, do a Copy Out of the file. The user selects a file, then clicks on the Check Out button, and the Check Out window will be displayed. This window is identical to the Copy Out window except for the title. The user selects the directory/file where the file will be written and clicks on the OK button. While the check-out is in progress an hourglass will be displayed and the window will be frozen. Note that if there is an existing file of the same name as the one to be checked out, the user will be notified and must verify the action.

The Cancel Check Out Function

This function cancels the check-out of a database file by unlocking the file so another user can check it out. The file is not removed from the user's work area. The user selects a file, then clicks on the Cancel Check Out button. While the operation is in progress an hourglass will be displayed and the window will be frozen.

The Check In Function

This function copies a file from the user's work area to the database. The selected file must have been checked out by the user using the Check Out function. The user selects a file, then clicks on the Check In button, and the Check In window will be displayed. This window is identical to the Copy Out window except for the title. The user selects the directory/file of the file will be read and clicks on the OK button. Note that the file names on the Check In menu must match the one on the Main Dialog. While the check-in is in progress an hourglass will be displayed and the window will be frozen.

The Impound Function

This function copies a file from the user's work area to the database. The selected file must not exist in the database (i.e., it is a new file). The user selects a file, then clicks on the Impound button, and the Impound window will be displayed. This window is identical to the Copy Out window except for the title. The user selects the directory and file that will be read and clicks on the OK button. While the impound is in progress an hourglass will be displayed and the window will be frozen.

Chapter 5: The Log Utility

Chapter Content

Refer to this Chapter for:

- Introduction
- Adding a Log
- Querying the Database
- Deleting Logs

Introduction

The log tool allows the user to record small pieces of important information on a day to day basis. It also will generate a report of logs that pass a user defined query.

Project menu

Name field

Date field
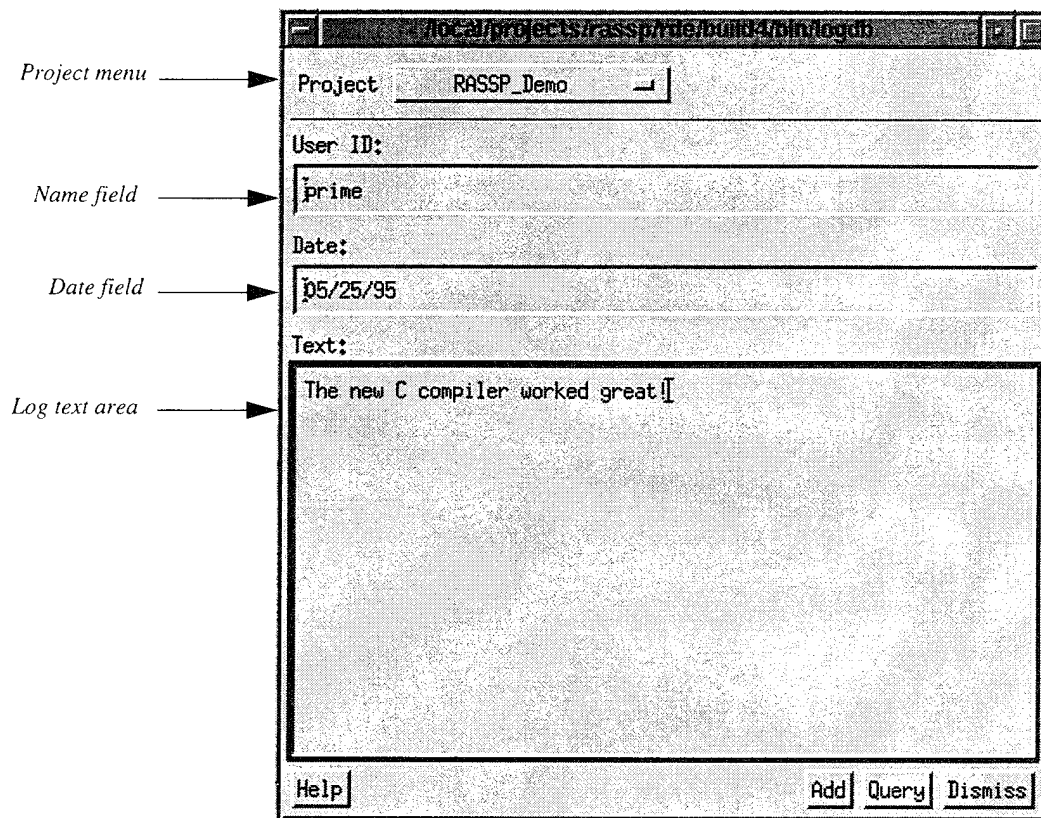
Log text area

FIGURE 1.   Log Tool Window

The main window consists of a project menu, a name and date field, a text area for entering logs, and a set of action buttons. You may quit the tool by pressing the *"dismiss button"* on the main window.

Adding a Log

1.   Choose a project by clicking the project menu and selecting the

project you wish to add your log to.

2. The users login name is shown by default. If the name field does not show the correct name, select the field and enter a new name.

3. Enter current date. The date must be typed in (MM/DD/YY) format.

4. Select the log text area and enter your log message.

5. Click the action button *"Add"* to add your log to the project log database.

Your new log is now stored in the database.

Querying the Database

The method of querying the log database is very simple. Only records which match the name field and the date field will be found. (see Figure 1, "Log Tool Window," on page 63) The text entered in the log text area when querying, must appear in the log or it will not be found. The symbol "*" may be use to indicate "do not care" when performing a query.

° **Name Field** - In the name field, enter the name of the user who entered the logs. This means only logs entered by that user will show up in the query results. A "*" may be entered if you do not care which user entered the log.

° **Date Field** - In the date field, enter a date range which you would like to search over.This date range must take the form: MM/DD/YY-MM/DD/YY. Either side or both sides of the separator may be "*" to have open ended date ranges or if you do not care about the date.

Examples:

° *-01/01/95 will return all records before 1995

° *-* will return all records

° 01/01/94-01/01/95 will return all records from 1994.

Enter a string which must appear in the log in the logs text area. Again a "*" will return all logs regardless of their content. If you enter *"software,"* then only logs containing the word "software" will show up in the query report.

```
╔══════════════════════════════════════════════════════╗
║                    Query Results                     ║
╟──────────────────────────────────────────────────────╢
║ ┌──────────────────────────────────────────────────┐ ║
║ │        Log DB Query Results                      │ ║
║ │ ──────────────────────────────────────────────── │ ║
║ │          Matching record count: 118              │ ║
║ │                                                  │ ║
║ │ Log  ID: 35                                      │ ║
║ │ User ID: falco                                   │ ║
║ │ Date   : 10/12/94                                │ ║
║ │ Text   : Debugging the Master/Slave protocol for the BFDB used on several │ ║
║ │ of the behavioral models.                        │ ║
║ │                                                  │ ║
║ │ Log  ID: 36                                      │ ║
║ │ User ID: falco                                   │ ║
║ │ Date   : 10/12/94                                │ ║
║ │ Text   : The handshake timing is working, I'm going to make sure the │ ║
║ │ data path is correct now.                        │ ║
║ │                                                  │ ║
║ │ Log  ID: 37                                      │ ║
║ │ User ID: falco                                   │ ║
║ │ Date   : 10/12/94                                │ ║
║ │ Text   : Simulator chokes when I try to view data array. Since, I'll be │ ║
║ │ doing an informal code review tommorrow I bring this up and see │ ║
║ │ if I can get a solution. In the mean time I'll start working on │ ║
║ │ the local bus logic.                             │ ║
║ │                                                  │ ║
║ │ Log  ID: 39                                      │ ║
║ │ User ID: falco                                   │ ║
║ │ Date   : 10/12/94                                │ ║
║ │ Text   : Working on the design for the Local Bus. I think I'll set it up so │ ║
║ └──────────────────────────────────────────────────┘ ║
║                                                      ║
║ ┌──────────────────┐   ┌──────────────────┐          ║
║ │  Write to file   │   │   Delete Logs    │          ║
║ └──────────────────┘   └──────────────────┘          ║
║ [I]              [ID1,ID2,ID3,...]            Dismis  ║
╚══════════════════════════════════════════════════════╝
```

FIGURE 2.   Query Results Window

The query results window consists of a results display area, write file button and field, and a delete log button and field. The results display area shows a list of all the logs which passed the query you defined. At the top there is a number showing the number of log records which did pass the query.

Saving Query Results

1.   Enter a file name in the field under the *"Write to file"* button.

2.   Press the *"Write to file button."*

All of the text in the display area has now been written to the file you named in a ASCII text format.

Deleting Logs

- In the field under the *"Delete Logs"* button you may enter a comma separated list of log identifiers. These identifiers are the "Log ID:" numbers which can bee seen on all logs in the results window.

- Press the *"Delete Logs"* button. The logs have been deleted from the database. You must rerun your query to see the updated list of logs.

Chapter 6: The Multi-Metric Browser Utility

Chapter Content

Refer to this Chapter for:

Introduction

The Multi-Metric Browser is a metric viewing tool. Via XRT widgets, it permits the user to view the metrics in different ways, including different metrics in a single graph. Metrics are retrieved from the shore database. The database is populated by the metric generators, the generators are normally run as a cron job. Multi-Metric Browser allows the addition of new metrics without any need to recompile the display system. New metrics can be added to the database using 'db_populator' utility. Users can specify project and subsystem and view metrics for them.

Available Metric Reports

There are nine major metric reports currently available and are accessible through the *"Multi-Metrics Browser."*

- **Code Size -** This report shows the lines of code for the user/time parametric. When the user parametric is chosen the metric report shows the lines of code generated by different users. When the time parametric is chosen the metric report shows the lines of code generated in different time periods.

- **New Open Problem -** This report shows the number of new open problems over a specified period of time. Its a good indicator of the organizations ability to respond to problems reported. This metric has time parametric, hence can be mapped with other metrics.

- **Total Number of Open Problems -** This report shows the total number of open problems that remain open at the end of the month.

- **Age of Open Problems -** This report classifies the list of open problems depending on their age. It could be used as an indication of responsiveness to customer needs.

- **Age of Closed Problems -** This report classifies the list of closed problems depending on the length of time taken to close them. It again is a good indication of the responsiveness to customer needs.

- **Phase Containment -** This report shows how well the design flaws are being captured in the earlier phase of development. Errors are defined as flaws discovered in a phase in which the flaw was inserted. Defects are defined as flaws discovered during a phase after the flaw was inserted.

- **Interface Complexity for C -** This report shows the interface com-

plexity of the various C files at locations specified in the 'proj_sub_dir' file. Its a multi-bar graph which shows the average interface count and the standard deviation (for the list of interface counts for the various modules in the file). In general, if the average count is low it implies the interface complexity is less. But if the standard deviation is high, then definitely there is some problem in terms of module complexity.

○   **Tool Usage** - This report shows the usage of various tools in seconds. There are two parametrics user and time. When the user parametric is chosen the metric report shows the number of seconds the tool was used by different users. When the time parametric is chosen the metrics report shows the number of seconds the tool was used during different time periods.

○   **Review Packets Created** - This report shows the number of review packets created for the user/time parametric. When the user parametric is chosen the metric report shows the number of review packets created by different users. When the time parametric is chosen the metric report shows the number of review packets created in different time periods.

To best illustrate how to use the multi-metric browser, refer to the information on the following pages. Figure 1, "Metric Browser StartUp Screen," on page 3 shows the Multi-Metric Browser as it comes up.
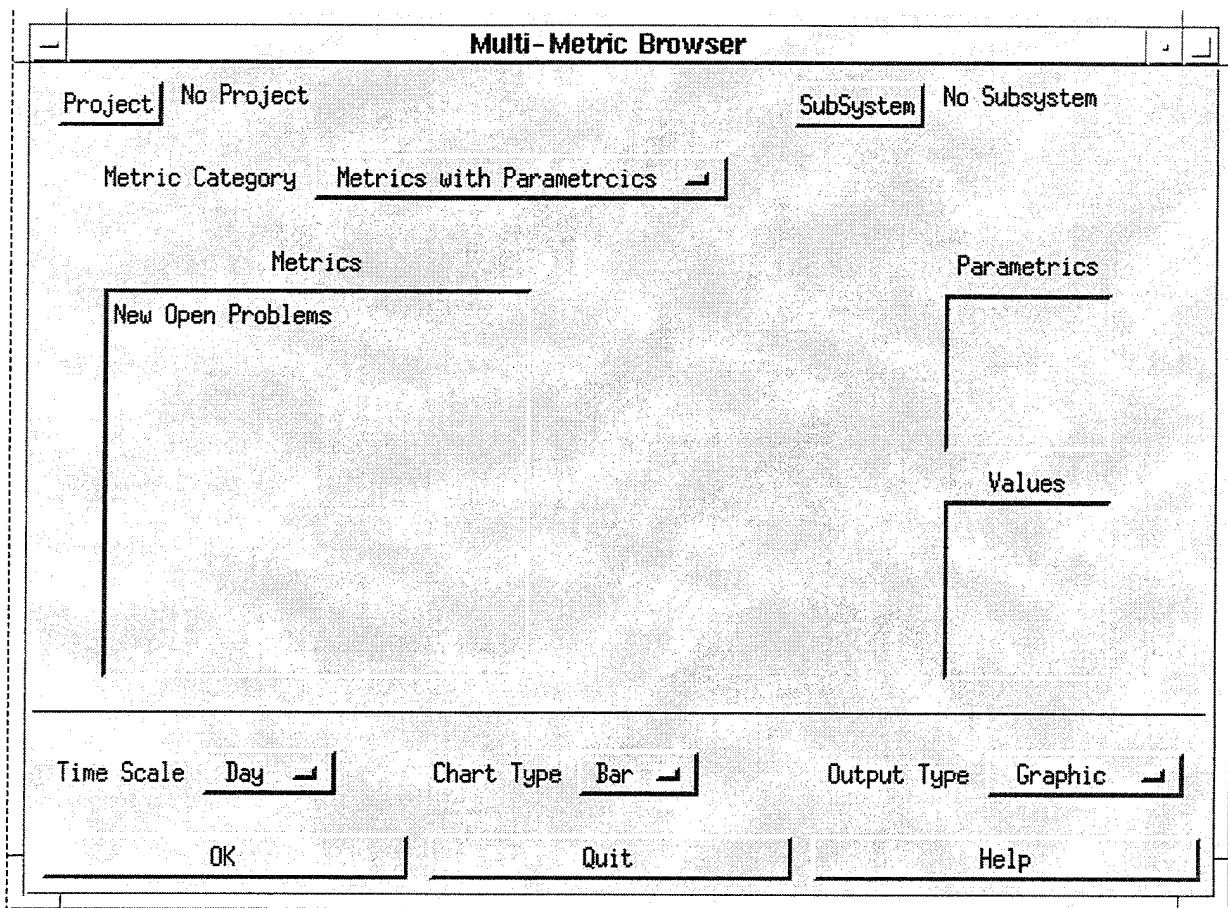
FIGURE 1.   Metric Browser StartUp Screen

Changing Metric Categories

There are two categories of metrics namely, those metrics that have parametrics and those that don't. Metrics with parametrics are those that can be viewed along with other metrics provided they have some common parametrics. Figure 2, "Metric Browser - Changing Metrics Category," on page 4 shows the case where the user has chosen a new category resulting in a new list of metrics on the metrics list.
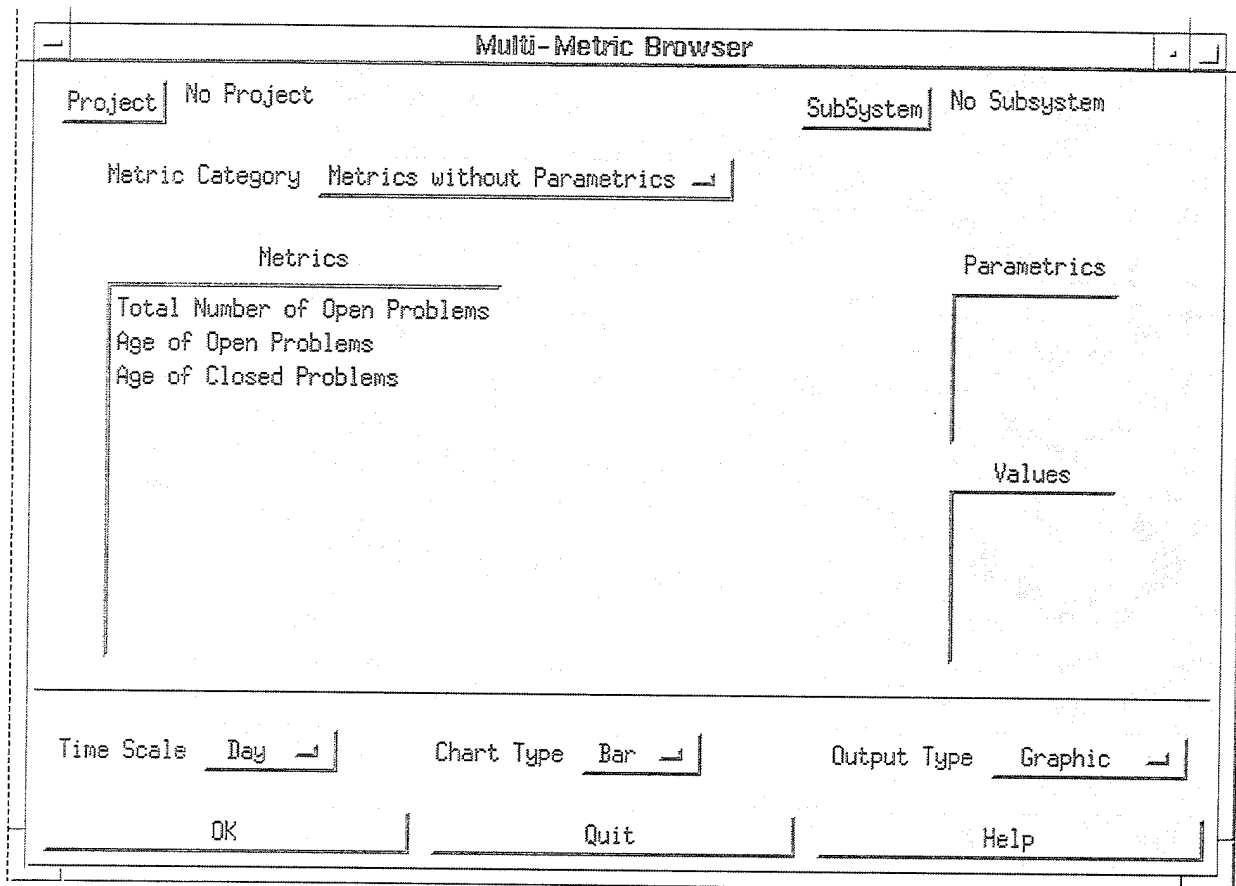
FIGURE 2.   Metric Browser - Changing Metrics Category

Changing Project and Subsystem

The project and subsystem buttons lets the user change the project and subsystem. Subsystem button lets the user choose from the list of subsystems available for the current project. Figure 3, "Project Selection Box," on page 5 shows the selection box that appears when the user clicks on the project button. When the user changes the project, list of metrics available gets updated too.
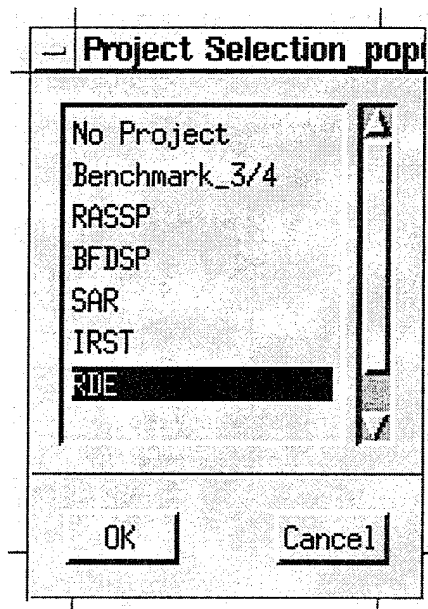
FIGURE 3.   Project Selection Box

Figure 4, "Metric Browser - New Set of Metrics," on page 6 shows the list of new metrics that are available for the new project selected. The list of metrics gets updated whenever the project or the subsystem or the metrics category is changed.
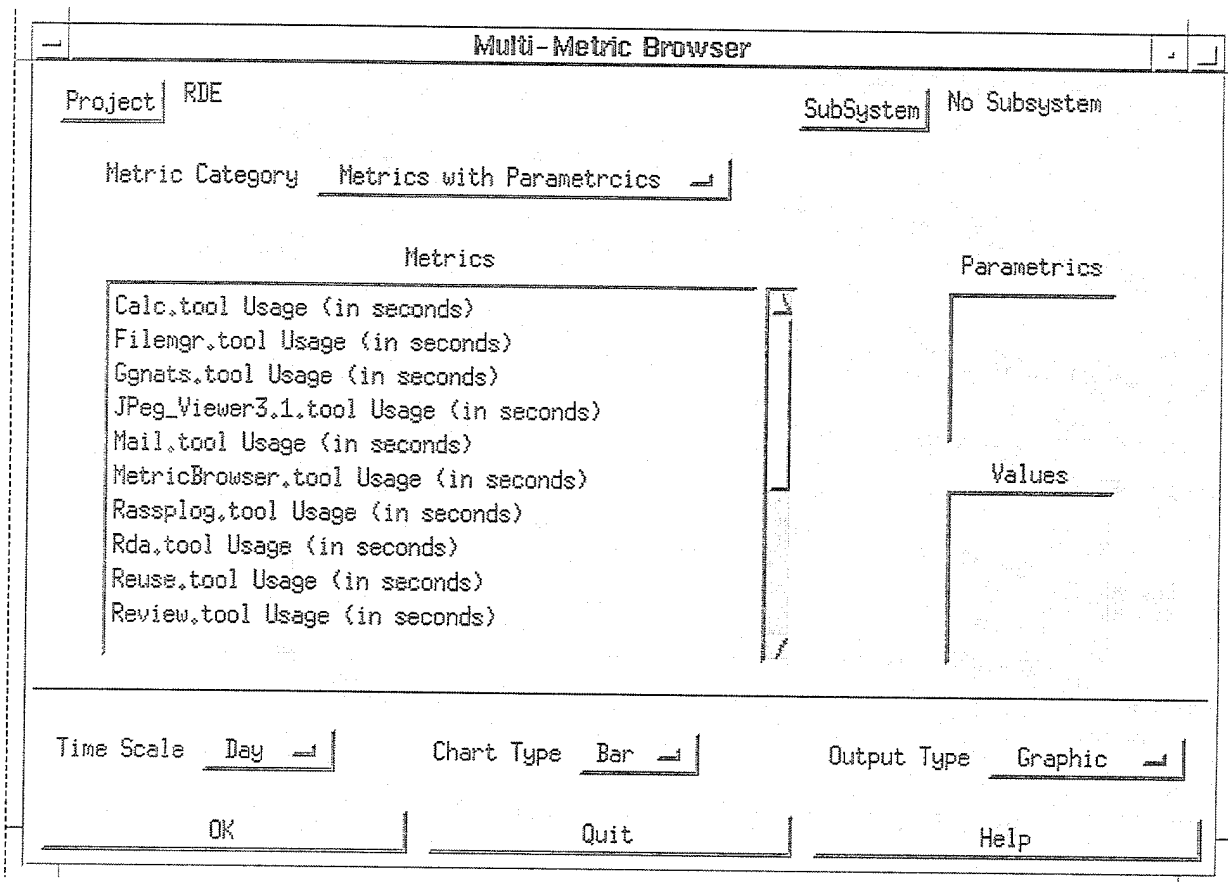
FIGURE 4. Metric Browser - New Set of Metrics

Getting Parametrics

When a metric is chosen from the metric list and if the metric has parametrics, the browser gets the list of parametrics associated with the selected metric. Figure 5, "Metric Browser - Getting Parametrics," on page 7 shows an example where the user has chosen the RDA tool usage metric and the browser is accessing the database to find the list of parametrics for the RDA tool usage metric.
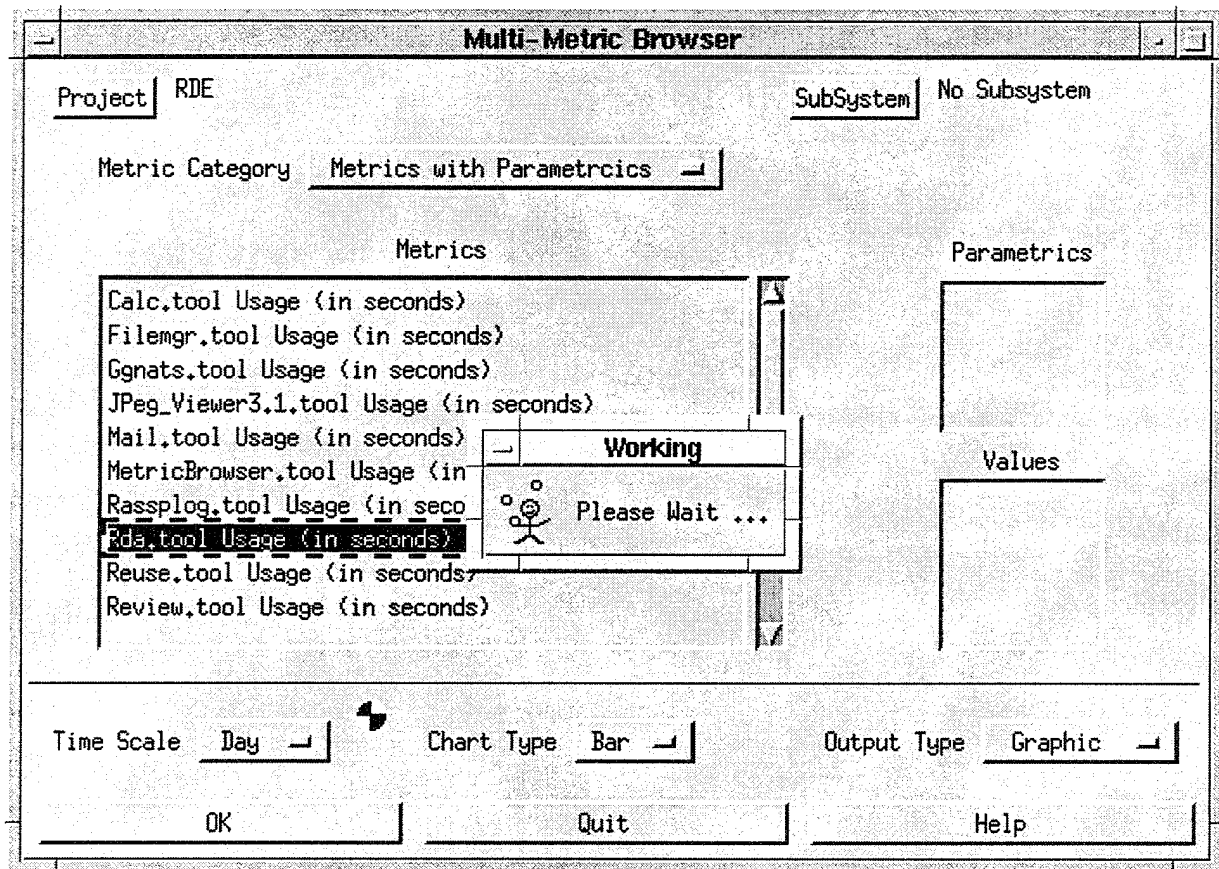
FIGURE 5. **Metric Browser - Getting Parametrics**

Accessing Database

Once the user has the list of parametrics, the user chooses a parametric from the list. When a parametric is selected, the browser accesses the database to find the parametric values for the chosen metric and parametric. Figure 6, "Metric Browser - Accessing Database," on page 8 shows an example where the user has chosen the user parametric and the browser is accessing the database to get the list of users who had used the RDA tool.
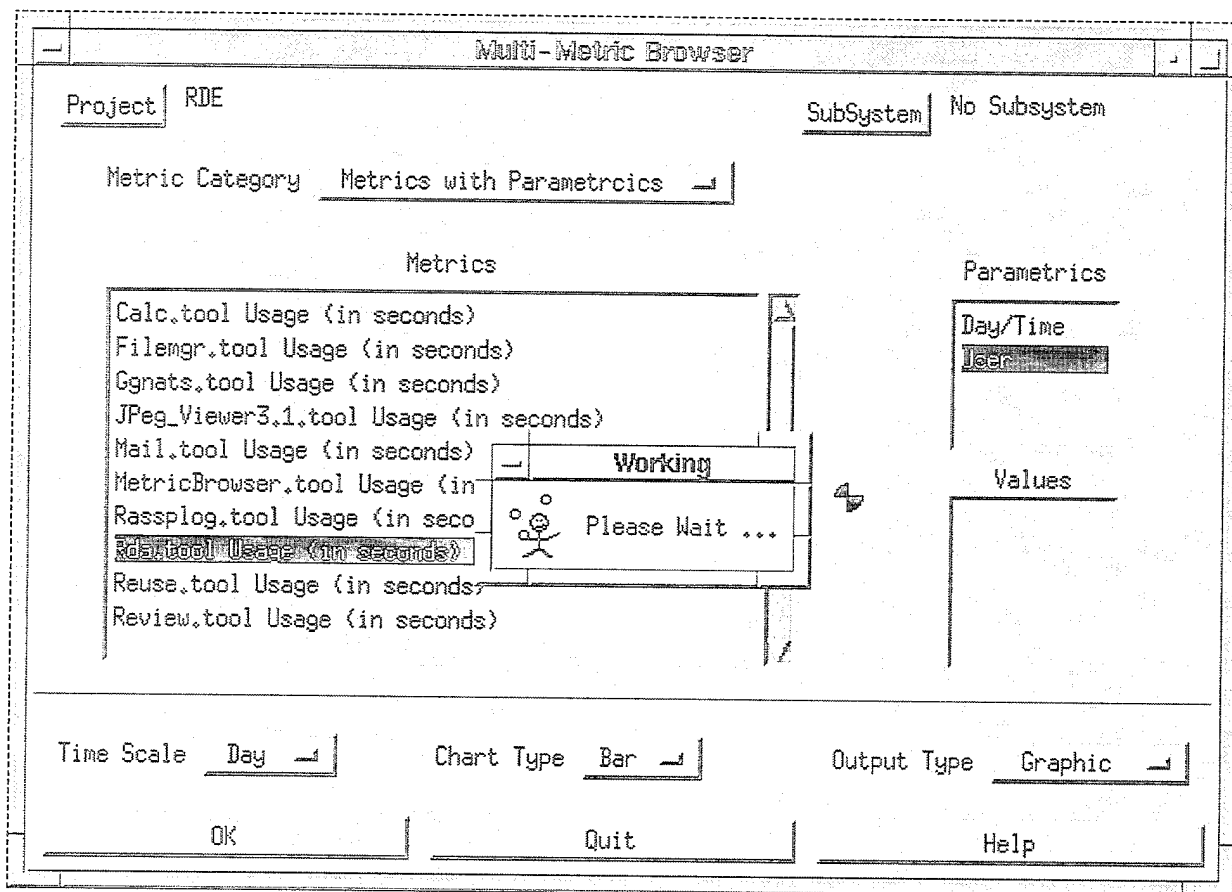
FIGURE 6.   Metric Browser - Accessing Database

Retrieved Values

The retrieved values for the selected parametric and metrics are displayed in the values list. Figure 7, "Metric Browser - Retrieved Values," on page 9   shows the list of users who had used RDA tool, or in other word the values for the user parametric in RDA tool usage metric
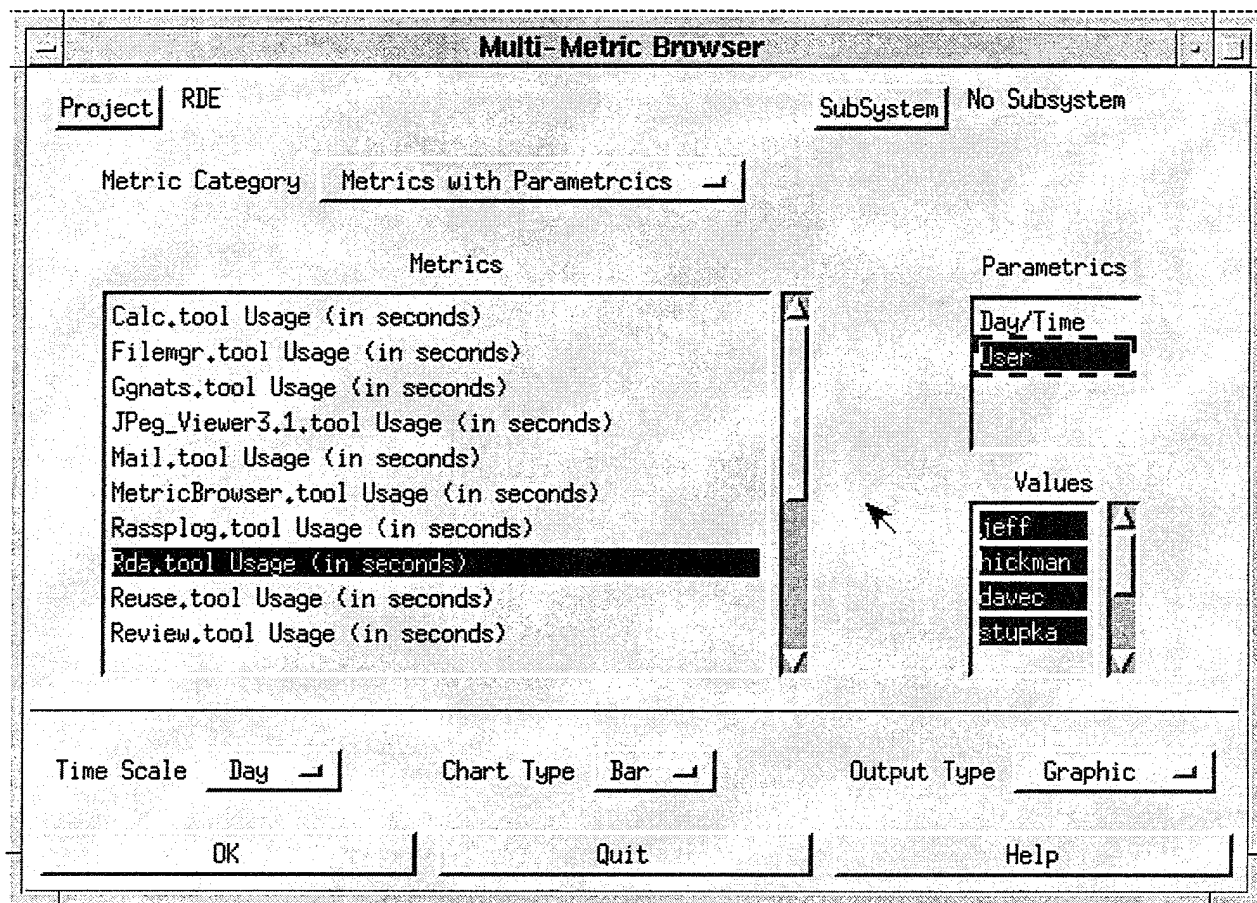
FIGURE 7.   Metric Browser - Retrieved Values

Common Values

When the user chooses another metric, list of parametrics is updated to show only the common parametrics. The value list is updated to include the list of values with atleast one of the selected metric defined. In the example case as shown in Figure 8, "Metric Browser - Common Values," on page 10 the user has chosen Reuse tool usage metric, as both are tool usage metrics the parametrics are the same. The value list is updated to have all the values for the selected parametric and both the metrics.
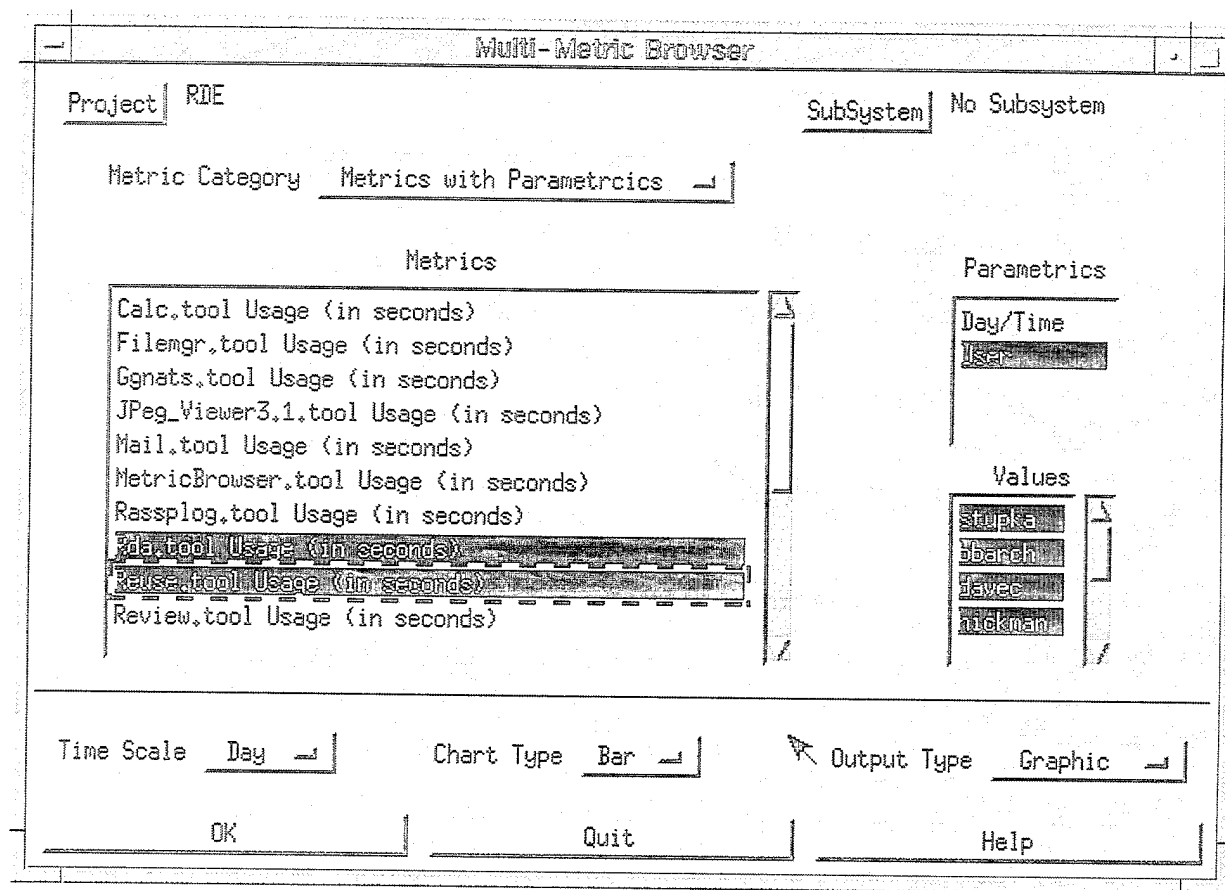
FIGURE 8.   Metric Browser - Common Values

Now when the user clicks on the ok-button the metric report is displayed. In Figure 9, "Metric Report - Bar Chart," on page 11 shows the data for the above example using a bar chart.
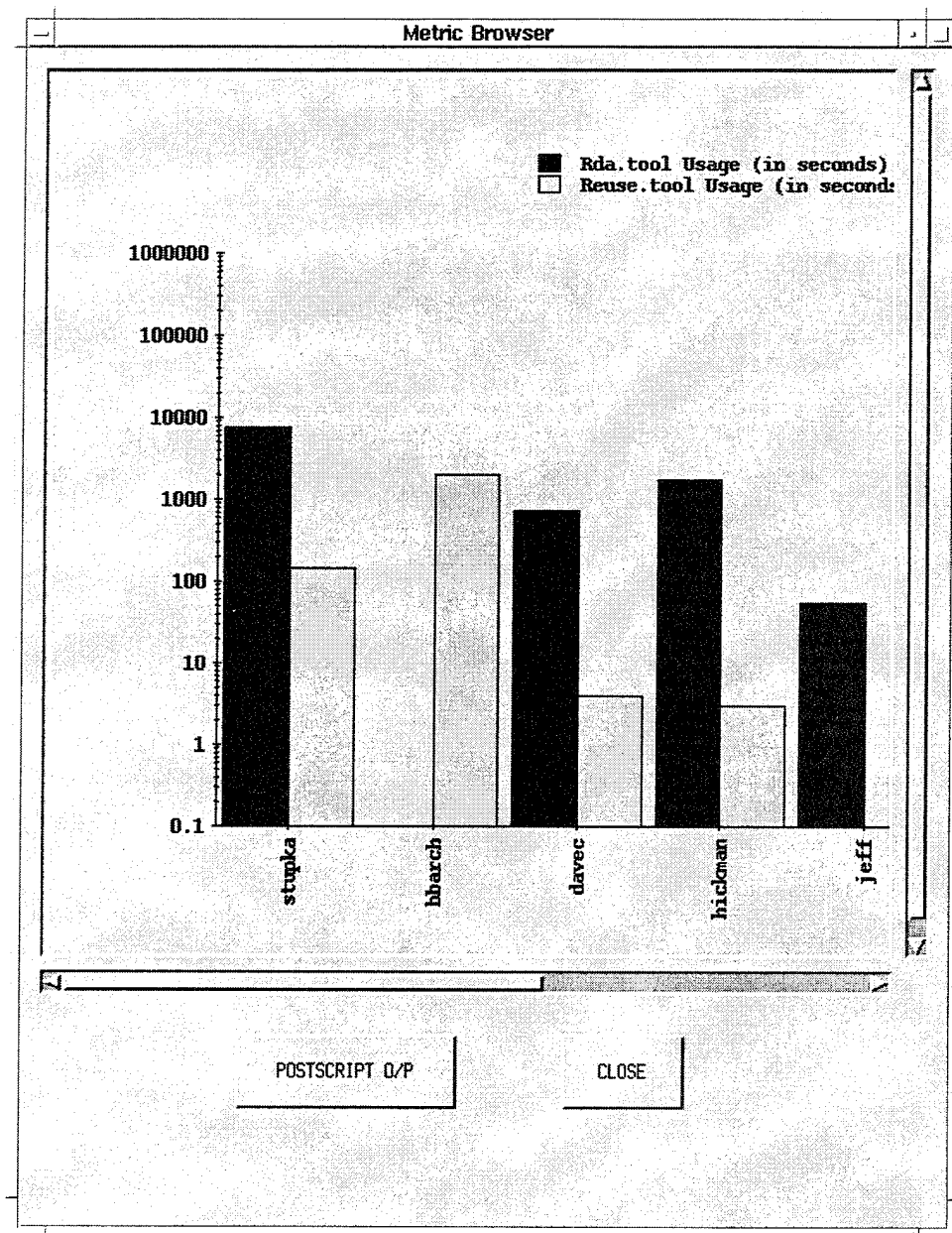
FIGURE 9.   Metric Report - Bar Chart

The user can also get a postscript dump of the graph by clicking on the POSTSCRIPT O/P button.

Chart Type Options

Metric browser supports three types of charts, namely bar, X-Y and line. The X-Y chart type is used to map two metrics. It maps one metric on the X-axis and another on the Y-axis. This chart can be used only when the metrics with parametrics category is chosen and there are exactly two selected metrics.Figure 10, "Metric Report - X-Y Chart," on page 12 shows an example of X-Y chart.
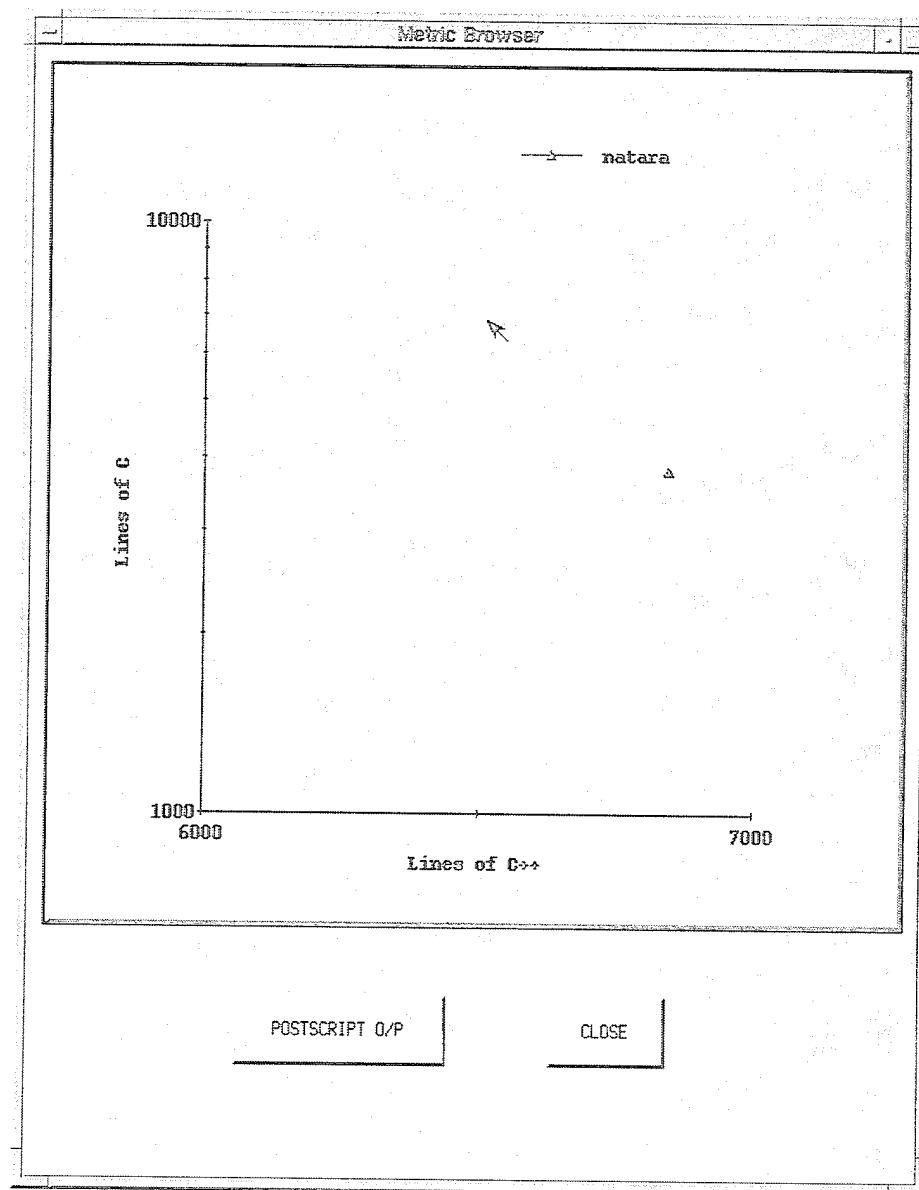
FIGURE 10.   Metric Report - X-Y Chart

Output Options

The browser supports three different output options. The metric data can be output as a graph or as a CSV (comma separated value) ascii file or as a postscript file. Postscript file option basically dumps the graph file into a postscript file. CSV output is chosen when the user needs to export the data on to a word or excel document. Figure 11, "Metric Output - Comma Separated Values," on page 13 shows the user choosing the CSV output option.
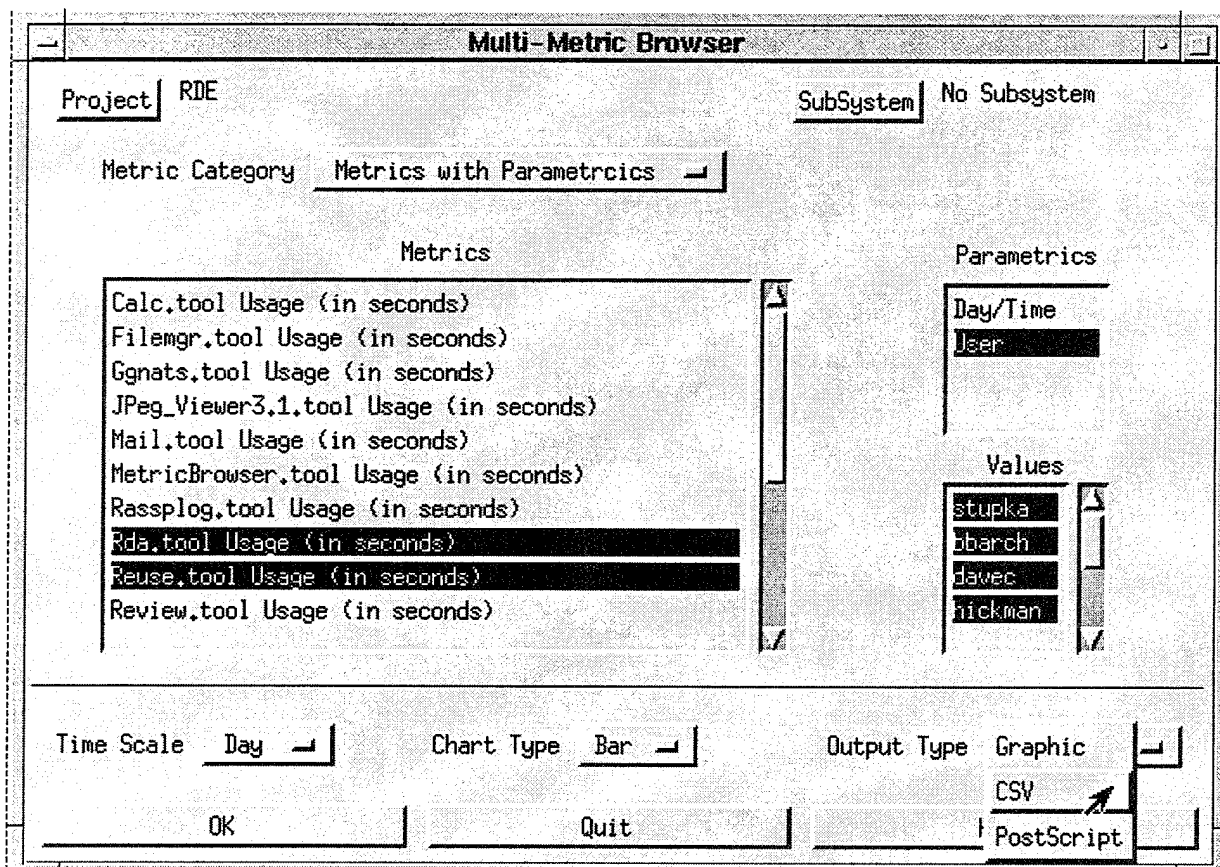
FIGURE 11.   Metric Output - Comma Separated Values

Figure 12, "Comma Separated Value File Selection Box," on page 14 shows the selection box that appears when the user chooses the CSV option and clicks on the ok-button. Now the user would be able to choose name and location for the CSV ascii file
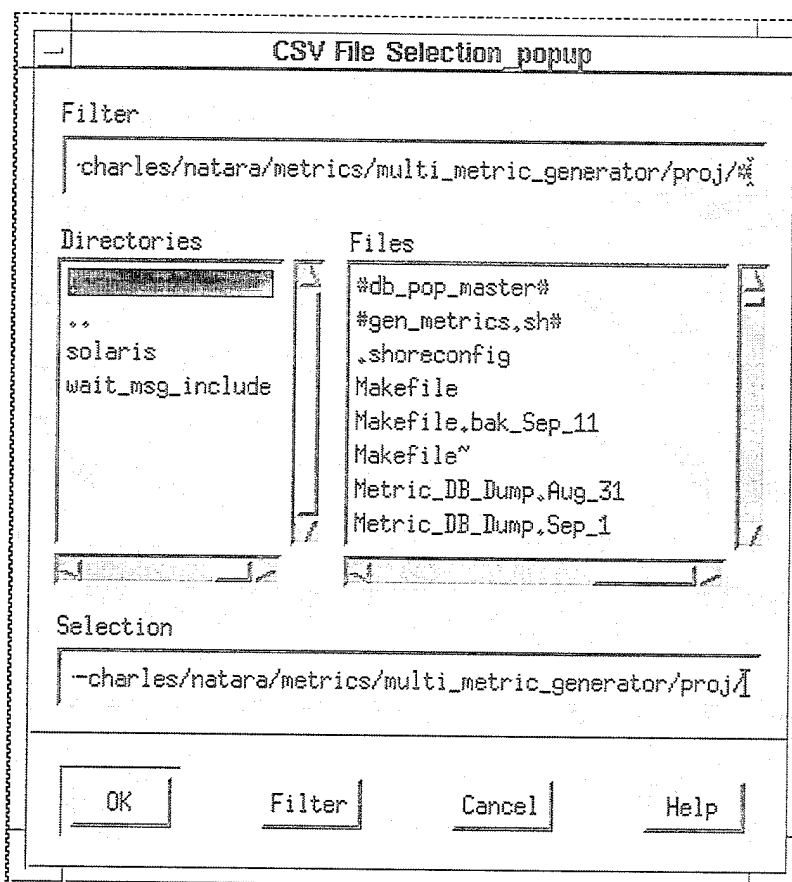
FIGURE 12.   Comma Separated Value File Selection Box

Database Populator   The metric browser allows the addition of new metrics without the need to recompile the system. This is accomplished by populating the database with the new metric in the 'right' way. The 'db_populator' utility at $RDE_HOME/bin helps populate the database. It takes in an ascii file. The ascii file has to be in a particular format. The format is shown in next page

```
<METRIC>
<PROJECT>project name</PROJECT>
<SUBSYSTEM>subsystem name</SUBSYSTEM>
<NAME>metric name</NAME>
<PARAMETRIC>
parametric 1
parametric 2
..... .....
parametric nn
</PARAMETRIC>
<X_LABEL>x axis label</X_LABEL>
<Y_LABEL>y axis label</Y_LABEL>
<LEGEND_LABELS>
legend label 1
legend label 2
.... ....
legend label n
<LEGEND_LABELS>
<DATA>
x_point_label 1, val11, val12, ..., val1n
x_point_label 2, val21, val22, ..., val2n
....
x_point_label m, valm1, valm2, ..., valmn
</DATA>
<DATA>
x_point_label 1, val11, val12, ..., val1n
x_point_label 2, val21, val22, ..., val2n
....
x_point_label m, valm1, valm2, ..., valmn
</DATA>
..... .....
<DATA>
x_point_label 1, val11, val12, ..., val1n
x_point_label 2, val21, val22, ..., val2n
....
x_point_label m, valm1, valm2, ..., valmn
</DATA>
</METRIC>
```

The format labels are described below

<METRIC> -- indicates the beginning of a metric

<PROJECT>..</PROJECT> -- used to specify the project name
for the metric, if there is none then
'No Project' is specified

<SUBSYSTEM>..</SUBSYSTEM> --used to specify the subsystem
name for the metric, if there
is none then 'No Subsystem'
is specified

<NAME>..</NAME> -- used to specify the name of the metric as
it appears in the metric list

<PARAMETRICS>
...
</PARAMETRICS> -- used to specify the parametrics, this would
be used to view the metric along with other
metrics that have common parametrics

<X_LABEL>..</X_LABEL> -- used to specify the x-axis label

<Y_LABEL>..</Y_LABEL> -- used to specify the y-axis label

<LEGEND_LABEL>
...
</LEGEND_LABEL> -- used to specify the list of legend labels

<DATA>
...
</DATA> -- used to specify one data set, there should one such for
every parametric. Each line corresponds to a data point,
it includes the point label and the values. There would
be one value for each legend label

</METRIC> -- indicates the end of the metric

Chapter 7: The Problem Report Utility

Chapter Content

Refer to this Chapter for:

• Introduction
• Conventions
• Concept
• Configuration
• Functionality
• Error Messages
• Support

Introduction

The Problem Report Tool, known as "ggnats", is a GUI, C++, version of the GNU GNATS "send-pr" Unix script. The GNATS system is the GNU Problem Report Management System (Version 3.2). GNATS is a bug-tracking tool designed for use with a central support site.

Conventions

The RDE Problem Report Tool, "ggnats", will be commonly referred to as: Problem Report Tool, ggnats (the name of the binary), and GG-NATS. In this context, GGNATS refers specifically to the problem reporting tool. Currently, the underlying problem management system and repository is still GNATS.

The GNU GNATS software will be commonly referred to as: GNATS, and gnats.

Concept

GGNATS and GNATS, with some other utilities, comprise an optional problem reporting mechanism that is provided with the RDE. This tool is used to submit problem reports for project or development related problems. This tool is being used extensively as the problem report management system for the RDE and other RASSP-related activities.

GNATS allows the support function to handle problems in a consistent and trackable manner through the use of email and a centralized problem report database.

At the support site, the problem reports are automatically assigned a tracking number and are tracked through resolution and closure by GNATS. An audit trail is automatically generated for each problem report. When a problem report is changed (e.g., priority change from low to medium) or its state changes (e.g., from "review" to "in-work"), the information is appended to the problem report. Feedback (via email) is provided to the requester each time a problem report is changed. Feedback about the entire problem database is provided by

the metrics reporting mechanisms.

Figure 1, "Problem Report Tool (Main Window)," on page 89 shows the main window for GGNATS, the Problem Reporting Tool. Each arrow indicates features explained in the following paragraphs.

Configuration

The Problem Report Tool reads default site information from a global RDE site configuration file. When "ggnats" is executed, the program reads necessary configuration information from the global defaults file. There is also an optional, local defaults file which allows users to customize their GGNATS configuration.

Global Default File

All users must reference the global defaults file, $RDE_HOME/etc/ggnatsGlobal, via the RDE_HOME environment variable (e.g. set via "setenv RDE_HOME <path_to_RDE_installation_directory>"). This file contain site-wide information including: the global problem category list, the global project (metric) phase list, the default GNATS repository site(s) and their respective email address(es).

The format of this global defaults file follows:

```
# Beginning of the ggnatsGlobal Configuration File for
# the RDE ggnats tool.
#
#
# Email information for generated problem reports.
TO_DEFAULT=RASSP;bugs@rassp.rassp.mot.com
#
# The default organization.
ORGANIZATION_DEFAULT=Motorola-GSTG
#
# The submitter-id for your site.
SUBMITTER_DEFAULT=kj
#
# The default GGNATS site.
GNATS_SITE_DEFAULT=kj
#
# The default mail address for PR submissions.
GNATS_ADDR_DEFAULT=
#
# The default release for this host.
RDE_GGNATS_RELEASE=RDEggnats V.1.0_1_B4, R.05.19.95
#
# Environment information.
SYSTEM_DEFAULT=SunOS kj 4.1.3_U1 2 sun4m
ARCHITECTURE_DEFAULT=sun4
MACHINE_DEFAULT=
#
# The ggnats local category list.
GGNATS_CATEGORY=Log Tool;Multi-Metrics Browser;Prob-
lem Report Tool;RDE Desktop;Remote Data Access
Tool;Reuse Tool;Review Tool;Other Tool...;Other Is-
sue...
#
# The Project Wide Phases
```

```
PROJECT_PHASE=Requirements Analysis;Preliminary De-
sign;Detailed Design;Coding and Unit Testing;Integra-
tion and Testing;Software System Testing;Production
and Deployment
#
#
# End of the ggnatsGlobal Configuration File
```

A hash symbol, "#", as the first character of the line, indicates a comment line.

Please refer to the comments above to explain the parameters. Refer to the GNATS documentation for additional information.

All of the configuration parameters should be defined in this file. The parameters critical to correct problem submission and tracking are: "TO_DEFAULT", "GGNATS_CATEGORY", "PROJECT_PHASE". "TO_DEFAULT" defines the email address of the GNATS administration site, "GGNATS_CATEGORY" contains the list of all categories for problem reports, and "PROJECT_PHASE" contains the list of all project phase names.

> *NOTE: These three parameters: "TO_DEFAULT",
> "GGNATS_CATEGORY", and "PROJECT_PHASE" will be overridden by the user's local defaults, if specified.*

Local Default File

The optional local user defaults file, ~<user_home_directory>/.ggnats, has the same format as the global defaults file, above, for the relevant configuration parameters. The local default file may contain the following parameters: "TO_DEFAULT", "ORIGINATOR_DEFAULT", "GGNATS_CATEGORY", "PROJECT_PHASE".

Menu Bar

Email Address

Submitter Name

Class Option Box
Priority Option Box
Severity Option Box

Category Button
Metrics Button

Synopsis

Description

Activities

Solution

Problem Report Tool

File   Edit   Options   Help

To: RASSP ⌐│ │bugs@rassp.rassp.mot.com

Submitter Name and Phone Number: │Dan Stupka x1395 H1175 pgr 4:

┌─Class?─────────┐   ┌─Priority?──────┐   ┌─Severity?──────┐
│ ✦ SW-bug   │   │ ✓ Low   │   │ ✓ Non-critical │
│   │   │   │   │   │
│ ✓ Doc-bug   │   │ ✦ Medium   │   │ ✦ Serious   │
│   │   │   │   │   │
│ ✓ Change Request│ │ ✓ High   │   │ ✓ Critical   │
│   │   │   │   │   │
│ ✓ Support   │   │   │   │   │
└────────────────┘   └────────────────┘   └────────────────┘

Category   **Metrics**

Enter a one line synopsis of the problem:

Enter a precise description of the problem:

Enter the code/input activities to reproduce the problem:

Enter the fix or workaround to solve the problem:

FIGURE 1.   Problem Report Tool (Main Window)

Functionality

GGNATS has straight forward simple functionality; GGNATS makes use of the suggested Motif menu and layout. Following is a description of each function the GGNATS provides.

Menu Bar

The menu bar supports the following menus. Refer to Figure 1, "Problem Report Tool (Main Window)," on page 89.

- **File:** This menu supports GGNATS file and exit functionality, and the problem submission capability. The "File" menu may be invoked with a mouse <click> on "File", or with the keyboard mnemonic <Meta-F>. Refer to the "File Menu" description below.

- **Edit:** This menu supports GGNATS edit functionality. The "Edit" menu may be invoked with a mouse <click> on "Edit", or with the keyboard mnemonic <Meta-E>. Refer to the "Edit Menu" description below.

- **Options:** This menu supports GGNATS options. The "Options" menu may be invoked with a mouse <click> on "Options", or with the keyboard mnemonic <Meta-O>. Refer to the "Options Menu" description below.

- **Help:** This menu supports GGNATS help functionality. The "Help" menu may be invoked with a mouse <click> on "Help", or with the keyboard mnemonic <Meta-H>. Refer to the "Help Menu" description below.

File Menu

This menu supports GGNATS file and exit functionality, and the problem submission capability. The "File" menu may be invoked with a mouse <click> on "File", or with the keyboard mnemonic <Meta-F>. The "File" menu supports the following menu options.

- **Open...:** This menu option may be invoked with a mouse <click> on "Open...", or with the keyboard mnemonic <O> with the "File" menu selected, or with the keyboard accelerator <Ctrl-O>.

  This option displays a Motif open dialog. Once the desired file has been selected, it will be imported into the current problem reporting session.

The file must be an ASCII text file in the "GGNATS" Problem Report Tool format created by the "Save..." menu option. "Open..." imports the following data: Category, Metrics, Problem Synopsis, Problem Description, Problem Input/Code Activities, Problem Fix/Workaround.

WARNING: If the file is not a GGNATS format file, it will not import correctly.

- **Open Description File...:** This menu option may be invoked with a mouse <click> on "Open Description File...", or with the keyboard mnemonic <D> with the "File" menu selected, or with the keyboard accelerator <Ctrl-D>.

  This option displays a Motif open dialog. Once the desired file has been selected, it will be imported into the current problem reporting session.

  This option opens the specified ASCII text file and imports the Problem Description information. This option has been added to support the command line "-filename" option from this GUI.

  The file must be an ASCII text file. "Open Description File..." imports the following data: Problem Description.

  WARNING: If the file is not an ASCII text file, it will not import correctly.

○ **Save...:** This menu option may be invoked with a mouse <click> on "Save...", or with the keyboard mnemonic <S> with the "File" menu selected, or with the keyboard accelerator <Ctrl-S>.

This option displays a Motif save dialog. Once the desired file-name has been selected, it will be saved/exported to a GGNATS formatted ASCII text file.

The file produced by the "Save..." menu option is suitable to be imported into a problem reporting session via the "Open..." option.

The following data is saved: Category, Metrics, Problem Synopsis, Problem Description, Problem Input/Code Activities, Problem Fix/Workaround. The file may be edited with a third party editor or word processor; however, you must maintain the order and format integrity of the file to insure a successful file import.

NOTE: If using another program to edit this file, it must be saved as an ASCII text file, or it will not import correctly.

○ **Submit Problem Report:** This menu option may be invoked with a mouse <click> on "Submit Problem Report", or with the keyboard mnemonic <R> with the "File" menu selected, or with the keyboard accelerator <Ctrl-R>.

This option submits the current report to the GNATS site specified in the "To:" ("Project:") field, or "Mail Address", refer to Figure 1, "Problem Report Tool (Main Window)," on page 89. The Problem Report is sent via the UNIX "mail" utility.

An information dialog will be displayed to inform the user of the status of the problem submission request.

NOTE: The Unix "mail" utility must be in the user's current path. If assistance is required then a local LAN administrator or RDE representative should be consulted.

- **Print Problem Report...:** This menu option may be invoked with a mouse <click> on "Print Problem Report...", or with the keyboard mnemonic <P> with the "File" menu selected, or with the keyboard accelerator <Ctrl-P>.

  This option displays a Motif print dialog. Once the desired print command has been specified, the default is "lpr", and the "Print" button selected, the current problem report information will be sent to the printer.

  The file must be an ASCII text file in the "GGNATS" Problem Report Tool format created by the "Save..." menu option. "Open..." imports the following data: Category, Metrics, Problem Synopsis, Problem Description, Problem Input/Code Activities, Problem Fix/Workaround.

  An information dialog will be displayed to inform the user of the status of the problem print request.

  NOTE: The Unix "lpr" utility, or other desired print utilities, must be in the user's current path. If assistance is required then a local LAN administrator or RDE representative should be consulted.

- **Exit:** This menu option may be invoked with a mouse <click> on "Exit", or with the keyboard mnemonic <X> with the "File" menu selected, or with the keyboard accelerator <Ctrl-X>.

  This option exits the Problem Report Tool immediately.

  WARNING: The Problem Report Tool (Version: RDEggnats V.1.0_1_B4, 05/24/95, obtained by invoking "ggnats -v") will not prompt you to send or save the current problem report if you select "Exit".

Edit Menu

This menu supports GGNATS edit functionality. The "Edit" menu may be invoked with a mouse <click> on "Edit", or with the keyboard mnemonic <Meta-E>. The "Edit" menu supports the following menu options.

○ **Clear All:** This menu option may be invoked with a mouse <click> on "Clear All", or with the keyboard mnemonic <C> with the "Edit" menu selected, or with the keyboard accelerator <Ctrl-C>.

This option clears information from the current problem reporting session. "Clear All" clears the following data: Category, Metrics, Problem Synopsis, Problem Description, Problem Input/Code Activities, Problem Fix/Workaround.

NOTE: Once "Clear All" has been selected, the cleared information can not be retrieved unless it was previously saved.

Options Menu

This menu supports GGNATS options. The "Options" menu may be invoked with a mouse <click>, or with the keyboard mnemonic <Meta-O>. The "Options" menu supports the following menu options.

○ **Category...:** This menu option may be invoked with a mouse <click> on "Category...", or with the keyboard mnemonic <C> with the "Options" menu selected.

This option displays a Motif "Problem Report Category" selection dialog. Once the desired category has been selected and the "OK" button <clicked>, the desired category will be placed in the text box next to the "Category" button.

The "Problem Report Category" dialog may also be selected by <clicking> the "Category" button on the main windows of the Problem Report Tool. Refer to Figure 2, "Problem Report Category Dialog," on page 99

NOTE: A category must be selected (i.e. displayed in the text box next to the "Category" button) in order to submit a problem report. The user will be warned appropriately if this situation occurs.

- **Metrics...:** This menu option may be invoked with a mouse <click> on "Metrics...", or with the keyboard mnemonic <M> with the "Options" menu selected.

    This option displays a Motif "Metrics Collection" selection dialog. Once the desired "Problem Inserted" and "Problem Detected" phases have been selected and the "OK" button <clicked>, the desired phase containment metrics are set. The phase containment metrics can be verified easily by selecting the "Metrics..." option again and viewing the dialog contents.

    The "Metrics Collection" dialog may also be selected by <clicking> the "Metrics" button on the main windows of the Problem Report Tool. Refer to Figure 3, "Metrics Collection Dialog," on page 99.

    NOTE: Phase containment metrics are not required in order to submit a problem report.

Help Menu

This menu supports GGNATS help functionality. The "Help" menu may be invoked with a mouse <click>, or with the keyboard mnemonic <Meta-H>. The "Help" menu supports the following menu options.

- **About Problem Report Tool...:** This menu option may be invoked with a mouse <click> on "About Problem Report Tool...", or with the keyboard mnemonic <A> with the "Help" menu selected.

    This option displays a Motif "About ggnats" dialog. This contains some brief information regarding GGNATS, the Problem Report Tool. When "OK" is <clicked> the dialog closes.

- **Help...:** This menu option may be invoked with a mouse <click> on "Help...", or with the keyboard mnemonic <H> with the "Help" menu selected, or with the keyboard accelerator <Ctrl-H>.

    This option displays a Motif "Problem Report Tool Help" window. This contains an on-line version of the GGNATS functionality. When "OK" is <clicked> the dialog closes.

Email Address

The email address via the "To:" ("Project:") popup menu. Multiple options may be specified. The user is allowed to choose from the available menu options only. If only one option is possible, that option will be the default. The user is "not" allowed to specify alternate sites from the GUI. Refer to Figure 1, "Problem Report Tool (Main Window)," on page 89.

Submitter Name

Enter the submitter name and phone number (including area code) in this field.

"Class?" Option Box

Select the appropriate class type. The choices are: Software bug, Document bug, Change Request, and Support.

- **Software bug:** This is the default and means a general product problem.

- **Document bug:** This is a problem with the documentation.

- **Change Request:** This is a request for a change in the way a program behaves.

- **Support:** This is a support problem or question.

"Priority?" Option Box

Select the appropriate priority type. The choices are: Low, Medium, and High. This selection will determine how quickly a problem is acted upon.

- **Low:** The problem should be solved in a future release.

- **Medium:** This is the default and means that the problem should be solved in the next release.

- **High:** A solution is needed as soon as possible.

"Severity?" Option Box

Select the appropriate severity type. The choices are: Non-critical, Serious, and Critical. This selection indicates the seriousness of the problem (i.e., missing functionality, etc.).

- **Critical:** The product, component or concept is completely non-operational or some essential functionality is missing. No work around is known.

- **Serious:** This is the default and means that the product, component or concept is not working properly or significant functionality is missing. Problems that would otherwise be considered "critical" are rated "serious" when a work around is known.

- **Non-Critical:** The product, component or concept work in general, but lacks features, has irritating behavior, does something wrong, or does not match its documentation.

Category Button

When the "Category" button is selected the Motif "Problem Report Category" selection dialog is displayed. Refer to the explanation above under "Options Menu" and "Category..."; also see Figure 2, "Problem Report Category Dialog," on page 99.

Metrics Button

When the "Metrics" button is selected the Motif "Metrics Collection" selection dialog is displayed. Refer to the explanation above under "Options Menu" and "Metrics..."; also see Figure 3, "Metrics Collection Dialog," on page 99.

Synopsis

Enter a One-Line Synopsis of the Problem. Enter a succinct one-line summary of the problem in the Synopsis Box.

Description

Enter a Detailed Description of the Problem. Enter a description of the problem within the Description Box. The more detailed the description, the easier it will be to diagnose and evaluate the problem.

Activities

Enter the Code/Input Activities needed to Reproduce the Problem - Enter the example code or input activities to reproduce the problem into the Activities Box.

Solution

Enter the Fix or Work-Around to Solve the Problem. Enter the fix or work-around for the problem used by the submitter, if any, into the Solution Box.

Error Messages

Appropriate error message dialogs will be displayed when error conditions are encountered. The user should note the error message, especially if it is problem that inhibits the use of the Problem Report Tool.

Support

The Problem Report Tool is an excellent mode for obtaining support on RDE services and tools, including GGNATS itself.

Issues and problems that require immediate attention should be initially addresses by your local LAN administrator or your local RDE representative.
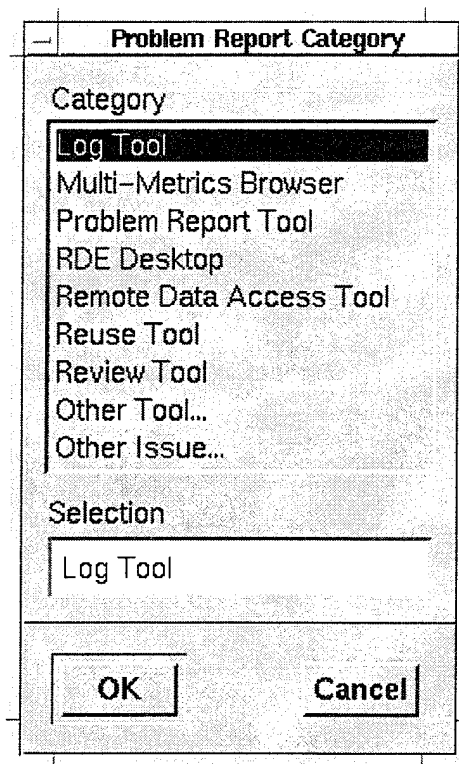
FIGURE 2. **Problem Report Category Dialog**
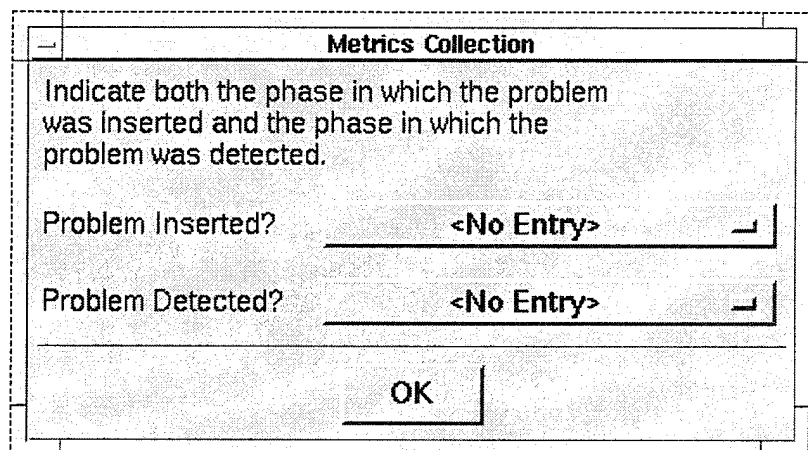
FIGURE 3. **Metrics Collection Dialog**

Chapter 8: The Reuse Utility

Chapter Content

Refer to this Chapter for:

• Introduction
• Reuse Configuration
• Reminders
• Functionality
• Command-Line Parameters
• Primary Functionality / Main Window
• The Add Application
• Querying the Database
• Displaying Matches
• Summary of Matches
• Other Dialogs

Introduction

In an effort to encourage reuse of functions and structures, the Reuse-Tool is designed to offer the user quick reference to software already available for reuse. The following gives a brief description of the applications which make up the Reuse Tool. It should be noted that this tool is built on and produced some reusable software.

Reuse Configuration

The reuse configuration file provides the path to the reuse directory, the programs with mailing lists only, and all the available languages. The information is used to create the language option menu, to keep track of who receives mail when changes occur to a record from a given project, and to set the path used by the file selection boxes. The program option menu is created with a list of projects found in a file used by the RDE.
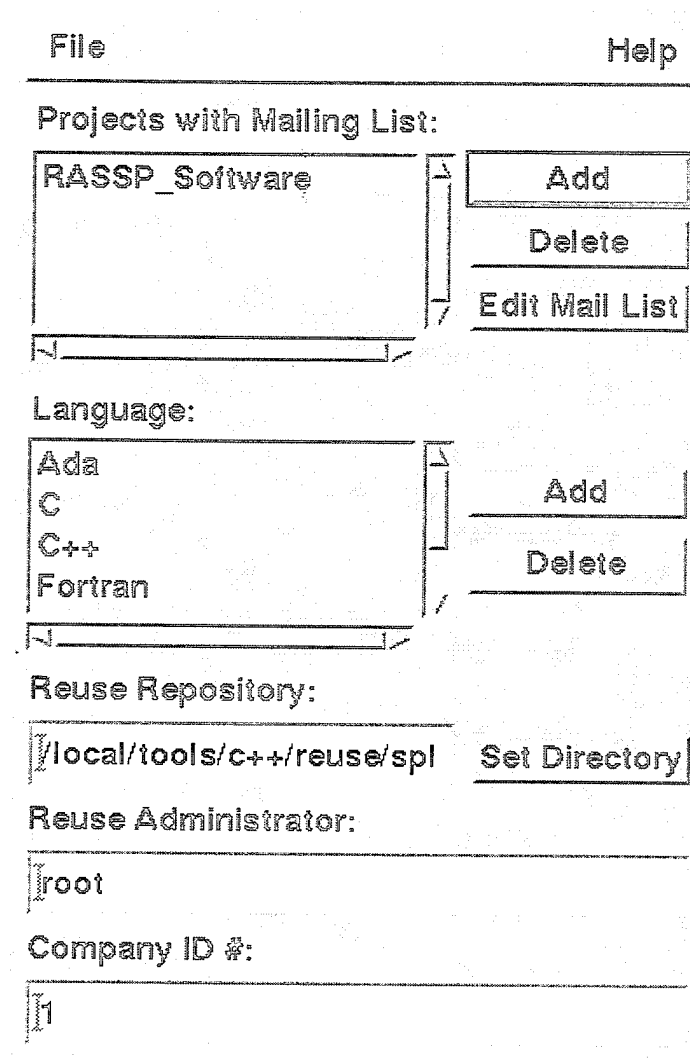
FIGURE 1.   Main GUI for the configuration file edits.

The reuse.cfg file should be edited through the command line option
'-config', that is, 'ReuseTool -config'. This pops up the configuration
main gui.(see Figure 1, "Main GUI for the configuration file edits.,"
on page 102) The menu bar 'File' menu contains two options: Save
and Quit. A help file is displayed upon the selection of the Help but-
ton in the Help menu.

To add a project to the list of projects with mailing list, press the Add
button next to the scrolled list under 'Projects...'. This will produce a
selection box.(see Figure 2, "Add project to projects with mailing
list.," on page 103) This will contain that same list of projects used
by the RDE and the program option menus in the Reuse Tool.
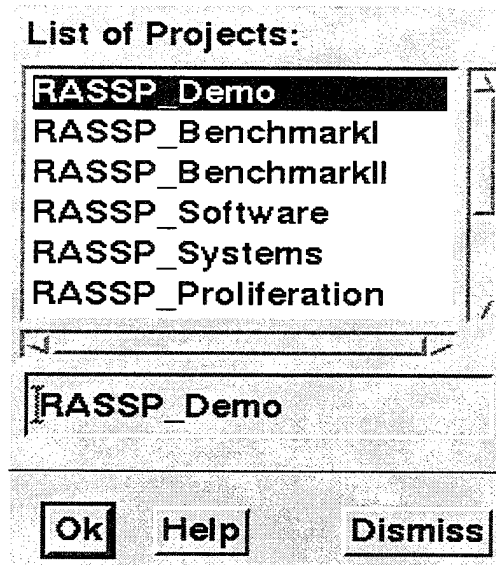
FIGURE 2.   Add project to projects with mailing list.

The Edit Mail List button allows the addition or removal of emailing addresses through a special gui.(see Figure 3, "Editing a projects mailing list, is done in this GUI.," on page 104)

> *NOTE: If a project exist in the main GUIs projects list, but does not have any mailing addresses, then it will be ignored.*

The Set Directory button pops up a directory browser.(see Figure 10, "Directory Browser," on page 114)

Mailing Lists:

super@sanders.com
teri@sanders.com

[Add]

[Remove]   [Dismiss]
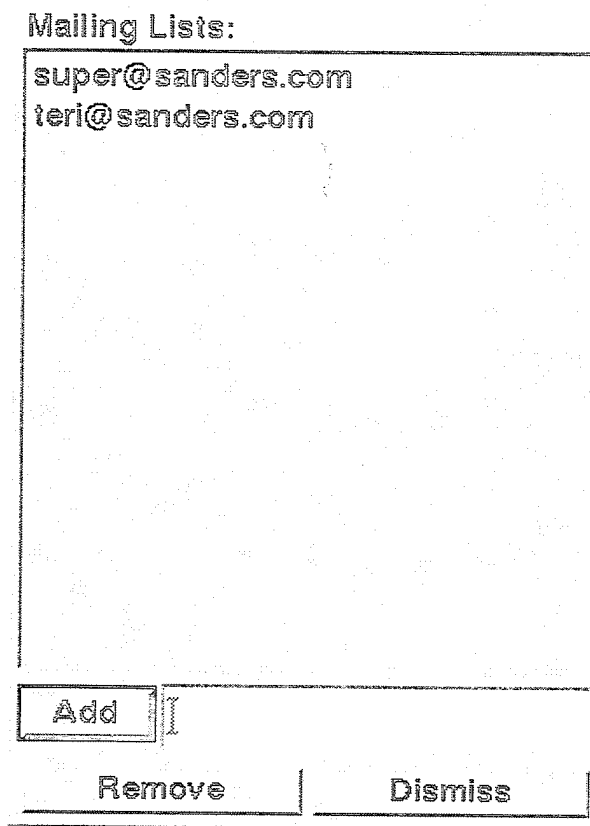
FIGURE 3.   Editing a projects mailing list, is done in this GUI.

An example of the reuse.cfg file:

```
NUM_PROGRAMS 2
 RASSP,john@rassp.company.com
 RDE,mike@rassp.company.com,etc@etc.com
NUM_LANGUAGES 5
 Ada
 C
 C++
 Fortran
 VHDL
PATH_REUSE
 /tmp_mnt/auto/asp_4b/tools/c++/reuse/splib/
COMPANY_CODE
 1
REUSE_ADMIN
 teri@mach.company.com <-- administrative user
END_PARAMETERS
```

Person(s) to mail to ——————▷

1 is Sanders
2 is Motorola
3 is Hughes ——————▷
4 is ISX

The address should be the user at the other company(s) who executes the ImportReuseMail executable. Preferably, this user is not a living, existing user, but rather a user made up for the sole purpose of running ImportReuseMail. That user should not receive mail other than ReuseTool generated mail, however, it will disregard any other mail, say group mail, if it arrives.The REUSE_ADMINistrator is a living user who can handle any permission, or configuration management problem discovered during the copying of files into the reuse directory. For example, if RCS or SCCS are used in the reuse directory, ImportReuseMail will try to write over an older copy of the file, checking it out if necessary. If it can not check out the file and the existing file is read only, then the new copy is put into a `.new' file and mail is sent to the administrator. It is the administrators responsibility to incorporate the new version of the file.The company codes are used by the databases to determine which company a record originated from.

Reminders

In addition to the reuse configuration file, you need to set the environment variable REUSE_CFG_DIR to the path containing the help files and the reuse configuration file. ReuseTool needs the resource file ReuseMainApp for both fonts and colors, as well as, translations which allow certain keystrokes within the text fields. The resource file should be at the path set to environment variable XAPPLRESDIR or in the same directory as the executable.To backup the records on the database, use the 'export' command line option with a filename. If the database needs to be filled with export data, then use the 'delete' option, followed by the 'import' option, with the exported file, should do the trick. Never import an export data file without running the 'delete' option. Also, never use another company's exported data file to import into your database. This causes problems with the generation of new IDs.

Functionality

Before running the Reuse Tool, there are some files that are expected by the tool; reuse.cfg, which contains the path to the reuse directory and list the available languages and programs with a mailing list. Applications within the reuse tool include: reuse_add_help.dat, help on the add application, reuse_dir_brw_help.dat, help on the directory browser, and reuse_config_help.dat, help on the configuration file. Also, the environment variable 'REUSE_CFG_DIR' should set to the path containing these files.

Command-Line Parameters

There are a few command line options associated with the Reuse Tool. They include the following.

metrics - displays metric information

v - displays version information

import - imports records from a file into database

export - puts a copy of all records from database to a file

delete - removes all records from database

config - edit the reuse.cfg file

For example:

```
ReuseTool -export <filename>
```

Primary Functionality / Main Window

The graphical user interface (GUI) for the main application of the Reuse Tool, offers three push button choices.(see Figure 4, "Main GUI for the Reuse Tool.," on page 107)

- ○ **Query** - Starts the query application, within which the reuse library/directory can be queried for reusable software structures; such as, functions and classes.

- ○ **Add** - Starts the add application, through which, all data is entered into the database for reusable structures.

- ○ **Quit Button** - Offers an ending to the Reuse Tool.

REUSE Tool Manager

Query

Add

Quit
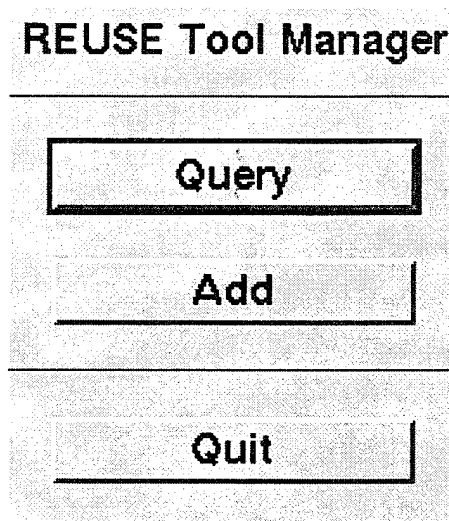
FIGURE 4. **Main GUI for the Reuse Tool.**

The Add Application

The add tool offers all the necessary fields for entering a function or a class into the reuse database. (see Figure 5, "Add Window," on page 109) In addition to the feature listed below, the project option menu will automatically show the 'current' project on the RDE bulletinboard, and if the user selects an other option from the menu, then it would be posted to that bulletinboard. The change would therefore be reflected by other RDE tools for that user.

The features of this tool are as follows:

- **Function/Class Name** - Expects the class name or functionality of the new item.

- **Description** - A more wordy description should be placed in this field.

- **Keywords** - Should include any common or standard words associated with the item. These are words from which you can query the item.

- **Libraries** - Libraries with full paths, are entered in the Libraries field.

- **Include Path** - The include files are divided into two fields. The full paths are placed in one field, followed by the file names in the file name field. The include files are usually header files when dealing with C/C++ structures.

○   **Add/View/Remove** - File names are added, viewed, and re-moved by utilizing the buttons to the right of the file name list.

Add File - The files which can be added, are those found in the reuse directory. Fields are also provided for a list of compilers, and the date associated with the structure. Option menus limit the possible languages and projects that may be chosen. These possibilities are controlled by the data in the configuration file. An example of how the structure may be used can be provided by pressing the code example button a menu pops up. The date and the creator field are set to the current date and the current user. The language and program menus default to the first item in each list or the last one selected from each list. Help is provided by pressing the appropriate button. Clear All clears all fields, except those which are given default values. The Add button commits all the data for a given item to the database and clears the fields for the next item.The Dismiss button ends the add application.

View File - Is used for a help file viewing text in the Add button, and by the View buttons in both the "Add" and the "Display of Matches" window. The View File is a scrollable window with a dismiss button attached. (see Figure 6, "View File - used for help files, viewing Text," on page 110)

Remove File - Allows user to remove a file.

○   **Show Code Examples** - Used for displaying the code examples. This is a scrollable window with a dismiss button, but it also contains a label identifying which class or function it is associated with. (see Figure 7, "Show Code Example," on page 110)

Function/ Class Name: spCosting class

Description: This class contains a number of functions and fields commonly used to calculate the cost of a project and various aspects of a project

Keywords: cost project budget

Libraries: /local/tools/accounting/libacc.a

Include Path: /local/tools/accounting/include

#include: budget_factors.h

File Names:
spTrace.cc
spTrace.h

Add

View

Remove

Program: RASSP_Demo

Date: 05/22/95

Language: C++

Compilers: CC2.1 g++

Creator: teri

Code Example

Add | Clear All | Help | Dismiss

FIGURE 5. Add Window

```
//////////////////////////////////// Help On Add Tool  ////////////

Function/Class Name :
        This field expects a character string.
        If the record is describing a class, then
        the class name would be entered.  Otherwise,
        the function, or general functionality,
        would be entered.  For examples:
        a class->        spBoolean
        a function->     binary_sort
```

                                                              Dismiss
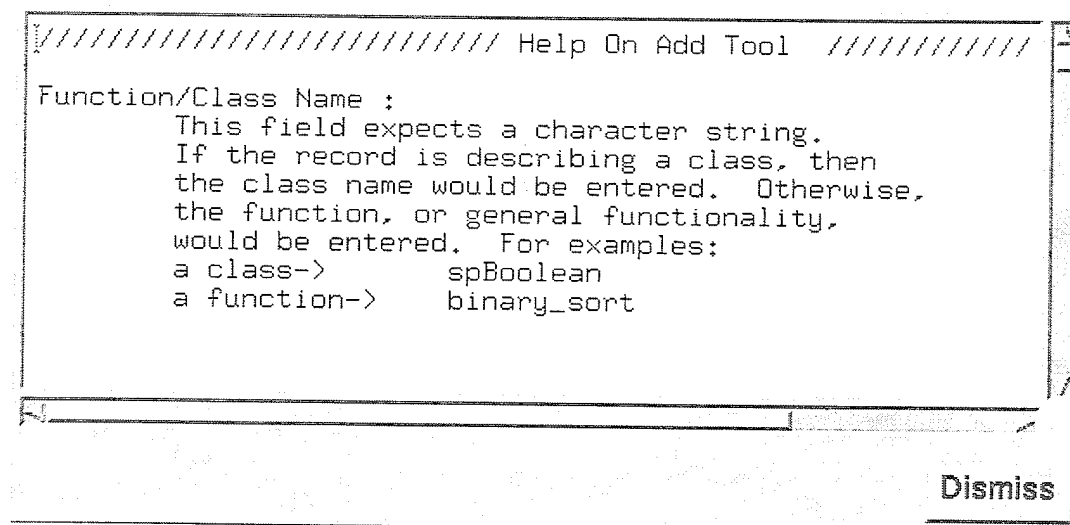
FIGURE 6.   View File - used for help files, viewing Text

Example for function: spBoolean

```
This type can be used in conditionals; such as, if-then,
while loops, and other loops.  For example:
        spBoolean fixed = FALSE;


        while (!fixed)
        {
            fixed = isItDone();
        }
where isItDone() returns an TRUE or FALSE.
```

                                                              Dismiss
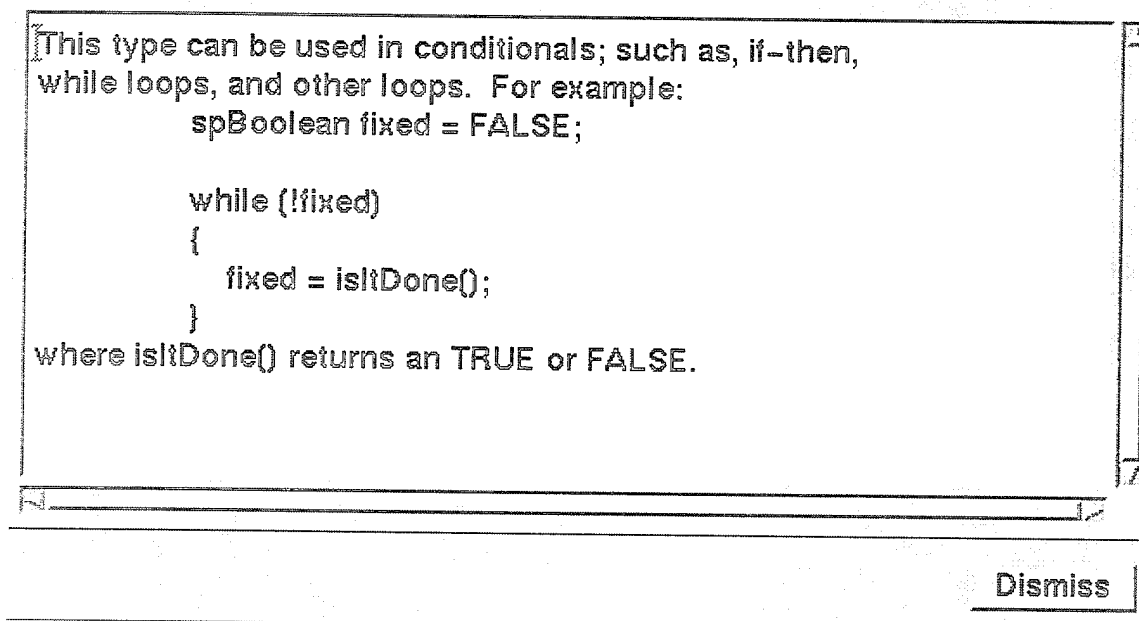
FIGURE 7.   Show Code Example

Querying the Database

Once the database has received entries for the current items in the re-use directory, the most useful application in the Reuse Tool is the query application. (see Figure 8, "Query display," on page 112)

- **Keyword Expression Field** - is the only field that requires an entry. This entry may either be a single keyword or language, or it may be a typical AND/OR expression. Some examples are:

```
C++ AND list OR display
(widget OR child) AND (display OR context)
vhdl OR (motif AND c++ OR ada)
list && link || node
tool OR aid
VHDL && Backup
C++ || Ada || Fortran || C
Motif
```

- **Language/ Programs** - if different than 'Any,' then from the matches found by the query, only those matching the language and/or program are given.

- **Creator/Date** - The same applies for the creator field and the to/from date fields. If these fields are not blank, then the matches found by the query must also match the data entered.

- **Query** - Once the expression is filled, and any optional fields, the *"Query"* button initiates the actual query. If no matches are found, then a message box pops ups. (See Figure 12, "'No matches found by query'," on page 115) If the expression failed the parsing routine, or was empty, then a parsing error message box pops ups. (see Figure 13, "'parse error'," on page 115) Otherwise, Figure 9, "Display Matches," on page 113 is produced, showing the first match.

- **Show Summary** - When pressed, will provide the summary of matches(seeFigure 11, "Summary of Matches," on page 115). Both the display and the summary are discussed below.

Filters

Creator: [

Languages: | Any ⌐ |

Date: [TO: [

Programs: Any   ⌐ |

Keyword/Query Expression: [

Query | Clear All | Show Summary | Dismiss |

FIGURE 8.   Query display

○ **Clear All/Dismiss** - Perform identical functions to their coun-
terparts in the add application.

Displaying Matches

If the query finds any matches, then the first one is automatically dis-
played in Figure 9, "Display Matches," on page 113. Very similar to
the add window in appearance, the display matches window shows all
the information for each item found by the query. Because the main
function of the Reuse Tool is to display the information about reus-
able software, this gui does not assume the user is interested in editing
the data; however if he is, then option is given through the 'Edit
Record' button.

○ **Arrow Buttons** - At the bottom of menu, allow the user to view
the next or previous item found by the query.

○ **Above the arrow buttons, record label** - Informs the user that
this record is nth record out of the total found.

○ **Copy** - The Copy button uses a directory browser application to
find the path to copy to, and that application uses the view file
for its Help button. A message application and a warning appli-
cation are used for messages and verification through out the
Reuse Tool. (see Figure 10, "Directory Browser," on page 114)

○ **Save Record** - Saves changes.

○ **Edit Record/Delete Record** - Using the Edit Record button fol-
lowed by the Delete Record button, removes the item from the
database.

- **Dismiss** - Ends the display application and clears the list of matches.

Function/ Class Name:	spOptionMenu class
Description:	This class is derived from the base class spWidget. It performs convenient functions for a motif option menu with the use of a class & objects.
Keywords:	gadget title pulldown motif widget
Libraries:	/local/tools/c++/reuse/splib/libSP.a
Include Path:	/local/tools/c++/reuse/splib
#include:	spPulldownMenu.h spWidget.h spOptionMenu.h

File Names: spOptionMenu.cc
spOptionMenu.h

Add
View
Remove
Copy

Program:	RDE	**Date:**	01/31/95
Language:	C++	**Compiler:**	g++ ccv2.0
Creator:	teri		**Show Code Example**

Record: 2 of 3   **Edit Record**   **Save Record**

◀ ▶   **Delete Record**   **Dismiss**

FIGURE 9.   Display Matches

Filter

/local/tools/c++/reuse/splib/

Directories   Files

.sentinel.tmp   ,spProjectsRetriever.cc
SCCS   ,spProjectsRetriever.h
pplus   –f
                     .make.state
                     .make.state+
                     .nse_depinfo
                     .pure
                     .purifyversion

Selection

/local/tools/c++/reuse/splib/

| OK | Filter | Cancel | Help |
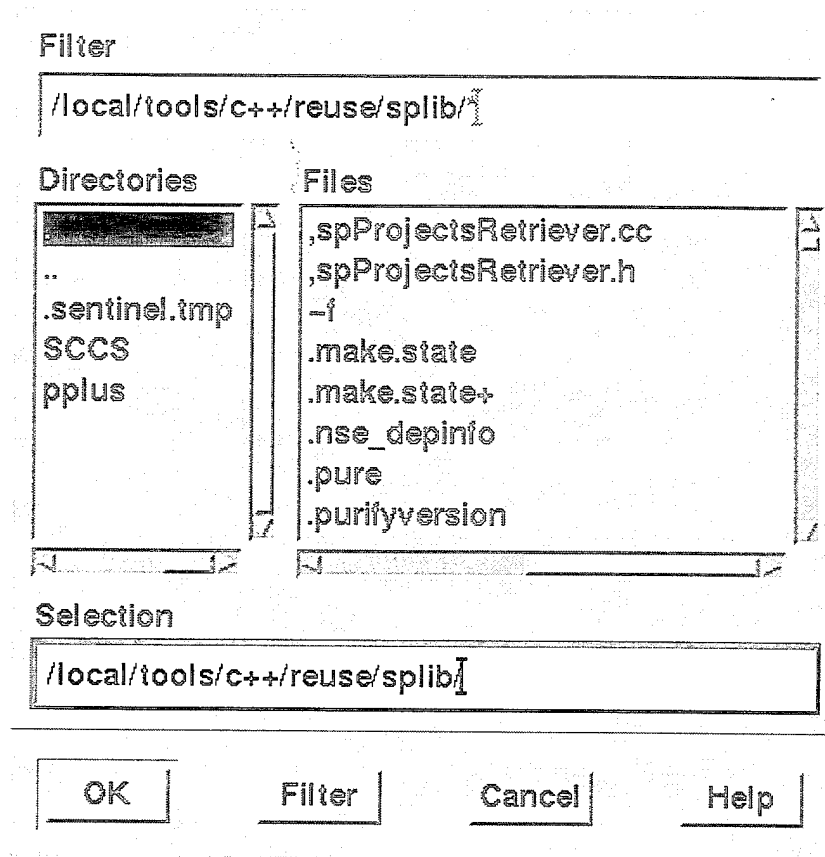
FIGURE 10.   Directory Browser

Summary of Matches

To activate a summary of the matches found by the query, click on Show Summary in Figure 8, "Query display," on page 112. Figure 11, "Summary of Matches," on page 115 pops up.

The function or class name, the language used, the date, and the first few words in the description for each match is displayed. By selecting an item from the summary list, the data for that item is immediately displayed in the display application. This is useful when dealing with a large list of matches. The Dismiss ends the summary application, but does not clear the list. Therefore, if the display application is still open, then the summary application can be re-displayed.

Function/Class	Language	Date	Description
spMotif help	C++	01/31/95	This is a group of helpful functions to
spOptionMenu class	C++	01/31/95	This class is derived from the base clas
spWidget class	C++	01/31/95	This class is representing a widget, as

Dismiss

FIGURE 11.   Summary of Matches
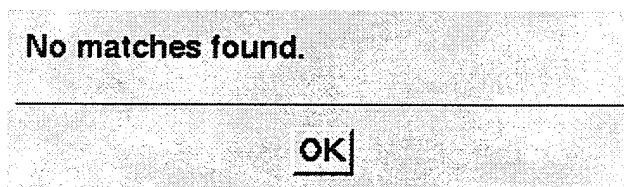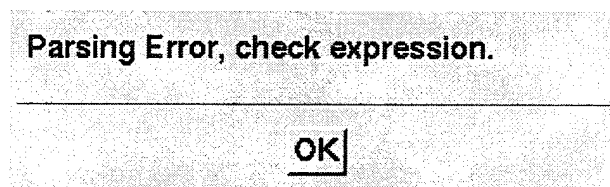
Other Dialogs

No matches found.

OK

FIGURE 12.   'No matches found by query'

Parsing Error, check expression.

OK

FIGURE 13.   'parse error'

Chapter 9: The Review Utility

Chapter Content

Refer to this Chapter for:

- Introduction
- Starting the Review Utility
- Creating / Starting a Review Package
- Reviewing a Package
- Comments
- Action Items
- Redlining a Design Object
- Additional Actions for Design Objects
- Closing a Review Package
- Stopping the Review
- Cancelling the Review
- Additional Features
- Reviewer Listing
- Named Package Filtering

Introduction

The Review Utility consists of a centralized server(Review-ToolServer) and multiple TCP/IP clients(ReviewTool). These tools help manage the peer review process by maintaining a database of all review information and actions.

Starting the Review Utility

The Review Utility client is started by launching it from the RDE DeskTop.

After starting the Review Utility client, the dialog shown in Figure 1, "Review Utility Startup Dialog," on page 119 is presented to the user.

The dialog is broken up into thirds. The top third contains the menu bar and information on the currently selected Review Package.

The middle third contains the list of available Review Packages with the current "Package List Type" setting (this setting can be either Reviewer, Submitter, Leader or All).

The bottom third shows the Submitter's comments for the currently selected Review Package (if applicable).

⊟	Review Tool	· ⌐

Review	**Package**	**Filter**	**Reviewer List**	**Options**	**Help**

Review Leader: Rob Hickman   Package Status: Closed

Package Submitter: Rob Hickman   Actual Close Date: 09/19/95

Reviewer Role(s):   Author, Review Leader, Hardware Engineer

Package Title	**Est. Close Date**	Reviewer Status
Review Pack #1	09/22/95	Completed
Review Pack #2	09/22/95	In Progress

Package List Type: **Reviewer** ⌐

Package Filter Title:

Submitter Comments/Instructions:

This is the first review pack.

FIGURE 1.   **Review Utility Startup Dialog**

The menu bar for the **Review Tool** dialog has the following entries:

- **Review** - Create, Start, Stop, Cancel, Reviewer List, Review Summary, Exit

- **Package** - Open, Close

- **Filter** - Show/Hide Named Filters, Named Filtering

- **Reviewer List** - Show/Hide Reviewer Lists

- **Options** - Show/Hide Message Center, Refresh

- **Help** - Context Sensitive Help, Help..., About Review Tool

Creating / Starting a Review Package

In order for a review to begin, a Review Package must be created and then started. There are two possible ways in which this may be done. A Review Package may be started at its creation or a Review Package may be started after its creation.

Creating a Review Package Without Starting

The following steps detail the process of creating a Review Package without starting it.

- With the "Package List Type" option menu set on any field, select "Review -> Create" from the menu bar. The dialog shown in Figure 2, "Create Review Package Without Starting," on page 122 will appear.

  NOTE: This figure shows all the fields completed.

- Select the appropriate value for the "Project" field. Note that this field is implemented as a combo box. Click on the down arrow to the right of the text field to display a drop down list of the available project values. Click on the desired project name to select it. This field is required.

- Select the appropriate value for the "Subsystem" field. As with the "Project" field, this field is implemented as a combo box. However, this field is inactive until a value for "Project" has been identified. This is because a subsystem must belong to a project. If a value has been selected for "Project" and the down arrow is clicked for "Subsystem", the drop down list that is displayed is a list of the available subsystems for the project specified in the "Project" field. Click on the desired subsystem name

to select it. Note that the "Subsystem" field is required.

- Fill in the appropriate values for the "Title" and "Review Leader" fields. It is not necessary to enter a value for the "Close Date" field because the Review Package is not being started (although it will contain the default value of three working days past the current date). The "Title" field specifies the title of the Review Package and therefore must be unique. The "Review Leader" field specifies the leader for this review. This value should be a valid user id for the review leader. Both the "Title" and "Review Leader" fields are required.

- Click the "Add" button under the Design Objects list box. A file browser dialog will appear. Click on all of the files that you wish to add as Design Objects to the Review Package. When you have finished selecting files, click on the file browser's "OK" button. This will copy all of the selected files to the Review Package. You may delete Design Objects from the Review Package by selecting the objects in the listing and then clicking the "Delete" button. Note that in order for a Review Package to be created, at least one Design Object must be specified.

- The Reviewer listbox is adjacent to the Design Object listbox. When creating a review package without starting it, there must be no Reviewers specified for the package.

  NOTE: If a default reviewer list has been specified for the current project/subsystem combination, the reviewers on the default list will be added to the Reviewer listbox automatically. If this is the case, select all of the Reviewers in the listbox and then click on the "Delete" button. When the confirmation dialog box pops up, click on the "Delete All" button to remove all of the Reviewers. Refer to the section, "Default Reviewer Lists", for more information regarding default reviewer lists.

- Use the "Submitter Comments" field to enter any instructions for the Review Package. These instructions will appear on the main Review Utility dialog (Figure 1, "Review Utility Startup Dialog," on page 119) when the Review Package is selected.

- Click on the "OK" button to create the Review Package. A dialog will appear informing you that the Review Package is being created.

After selecting "OK", the client application will submit the package to the server. The server will generate an email message and send it

to the user id identified as the Review Leader. This email message tells the leader that a package needs attention.

```
┌─────────────────────────── Create Review Package ───────────────────────────┐
│                                                                              │
│  Project:        │ RDE                                                    ▼│ │
│                                                                              │
│  Subsystem:      │ Review Utility                                         ▼│ │
│                                                                              │
│  Title:          │ Redliner Interface │     Review Leader: │ hickman │        │
│                                                                              │
│  Close Date:     │ 09/25/95                                                 │ │
│  (Month/Day/Year)                                                            │
│                                                                              │
│   ┌──── Design Objects ────┐         ┌── Reviewer Name ──┬── Reviewer Role ─┐│
│   │ DesignObject.C         │         │                   │                  ││
│   │ do_shell.C             │         │                   │                  ││
│   │ do_shell.H             │         │                   │                  ││
│   │ ColorSelectorDlg.cc    │         │                   │                  ││
│   │ redline.cc             │         │                   │                  ││
│   │ RedlineTool.cc         │         │                   │                  ││
│   │ shapeCommentPixmap.cc  │         │                   │                  ││
│   │                        │         │                   │                  ││
│   │                        │         │                   │                  ││
│   │                        │         │                   │                  ││
│   │    Add    │   Delete   │         │  Add  │  Delete  │  List            ││
│                                                                              │
│  Submitter Comments                                                          │
│  ┌────────────────────────────────────────────────────────────────────────┐│
│  │ Please review these design objects│                                     ││
│  └────────────────────────────────────────────────────────────────────────┘│
│                                                                              │
│      OK        │          Cancel          │          Help                   │
└──────────────────────────────────────────────────────────────────────────────┘
```

FIGURE 2.   Create Review Package Without Starting

Starting the Review Process After Creation

A Review Package that is created may only be started by the user who was identified as the Review Leader. The Review Leader is notified

by email that a Review Package has been created and now needs to be started. The Review Leader can modify the previously defined Review Package if necessary. These modifications are limited to specifying additional Design Objects.

To start a review package after it has been created, perform the following actions.

- Select the "Leader" filter option from the "Package List Type" option menu. A list of all packages for which the user is the defined leader will appear. (see Figure 3, "Review Utility With Review Package," on page 124 for an example).

- Highlight the Review Package in the listbox that was identified in the email notification and select "Review -> Start" from the menu bar.

```
┌─┐                          Review Tool                          ┌─┐┌─┐
│─│                                                               └─┘└─┘

  Review    Package    Filter    Reviewer List    Options                Help

  Review Leader:     Rob Hickman           Package Status:    Unopened

  Package Submitter:  Rob Hickman          Actual Close Date:  Not Applicable


  Reviewer Role(s):   Author, Review Leader


             Package Title              Est. Close Date        Reviewer Status

  Review Pack #1                        09/22/95             Completed
  Review Pack #2                        09/22/95             In Progress
  Redliner Interface                    None Identified      Never Opened



  Package List Type:    Leader

  Package Filter Title:


  Submitter Comments/Instructions:

  Please review these design objects.
```

FIGURE 3.   Review Utility With Review Package

- The dialog shown in Figure 4, "Start Review Dialog," on page 125 will appear:

FIGURE 4.   Start Review Dialog

- Fill in the appropriate value for the "Close Date" field. The "Close Date" field specifies the date by which all reviewers should be finished with reviewing the Design Objects in the Review Package. Note that this field has a default value of three working days past the current date. This field is required.

- Specify the Reviewers for this Review Package. Reviewer specification may be done through reviewer lists, individual re-

viewer specification or a combination of both.

Reviewer Specification with Reviewer Lists

A reviewer list provides a convenient method of specifying a regular group of reviewers for a review package. For a detailed description of reviewer lists, please refer to the section, "Default Reviewer Lists".

○ If a default reviewer list has been specified for the current project/subsystem combination, the reviewers on the default list will be added to the Reviewer listbox automatically. If you wish to use a reviewer list other than the default, click on the "List" button below the Reviewer listbox. The dialog presented in Figure 5, "Selecting an Alternate Default Reviewer List," on page 126 will appear.
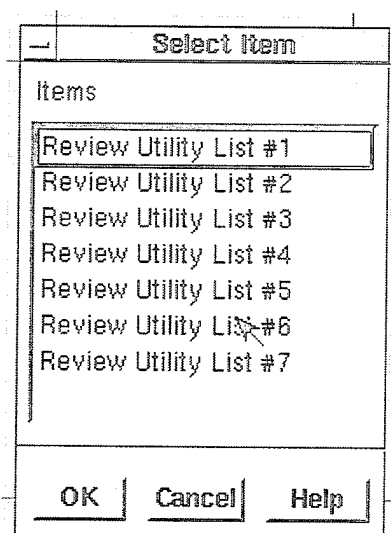
```
┌─┬──────────────────────────┬──┐
│─│        Select Item       │  │
├─┴──────────────────────────┴──┤
│ ┌──────────────────────────┐  │
│ │ Items                    │  │
│ │ ┌──────────────────────┐ │  │
│ │ │Review Utility List #1│ │  │
│ │ ├──────────────────────┤ │  │
│ │ │Review Utility List #2│ │  │
│ │ │Review Utility List #3│ │  │
│ │ │Review Utility List #4│ │  │
│ │ │Review Utility List #5│ │  │
│ │ │Review Utility List #6│ │  │
│ │ │Review Utility List #7│ │  │
│ │ │                      │ │  │
│ │ └──────────────────────┘ │  │
│ │                          │  │
│ │ ┌────┐ ┌──────┐ ┌──────┐ │  │
│ │ │ OK │ │Cancel│ │ Help │ │  │
│ │ └────┘ └──────┘ └──────┘ │  │
│ └──────────────────────────┘  │
└───────────────────────────────┘
```

FIGURE 5.   Selecting an Alternate Default Reviewer List

○ Select the name of the default reviewer list you wish to use for this review package. Click on the "OK" button to add the reviewers on the selected reviewer list to the Reviewer listbox.

Individual Reviewer Specification

You may also specify Reviewers for the Review Package on an individual basis.

- To add additional reviewers, click on the "Add" button below the Reviewer listbox. The dialog presented in Figure 6, "Add Reviewers Dialog," on page 127 will appear.
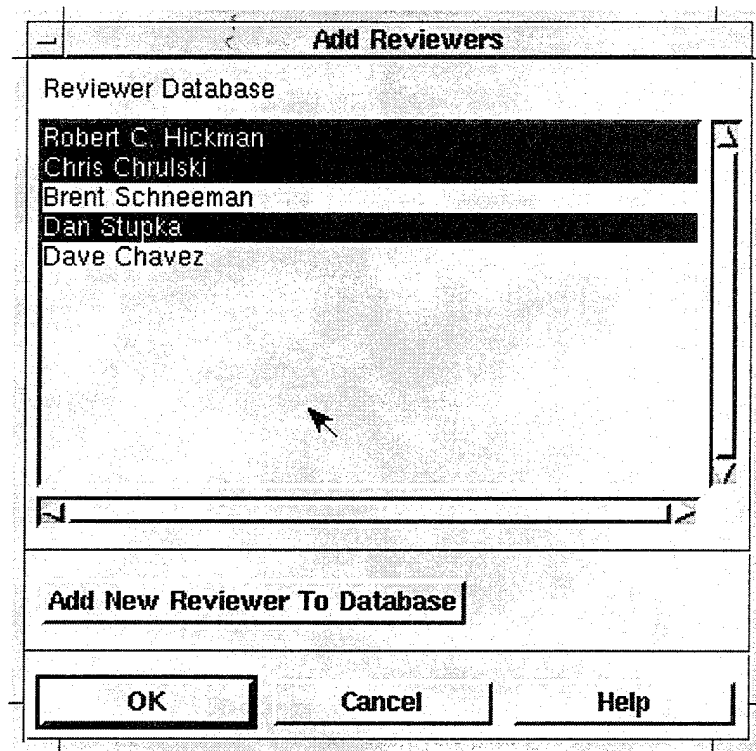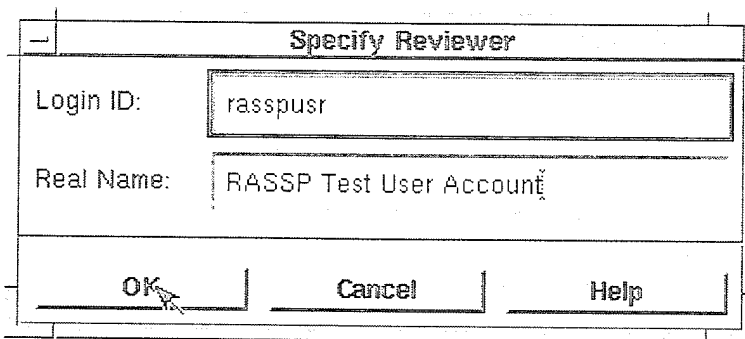

FIGURE 6.   Add Reviewers Dialog

- To add reviewers to the Review Package, click on the desired individual's name in the Reviewer Database listbox to highlight the name. If you wish to add a reviewer to the Review Package that does appear in the listing, click on the "Add New Reviewer to Database" button. When you do this, the dialog presented in Figure 7, "Specify Reviewer Dialog," on page 128 will appear.

FIGURE 7.   Specify Reviewer Dialog

○   To add the new reviewer to the database, fill in the values for the "Login ID" and "Real Name" fields. The "Login ID" should be a valid user id and the "Real Name" should be the full name of the user corresponding to the user id. After you have entered the values, click on the "OK" button. The new user will now be added to the list and you will then be able to select the reviewer name to add the reviewer to the Review Package.

○   After you have selected all of the individual reviewers you wish to add to the Review Package, click on the "OK" button. You will now be presented with the dialog shown in Figure 8, "Reviewer Role Specification Dialog," on page 129. This dialog is used to assign roles to all of the reviewers for the Review Package.

FIGURE 8.   Reviewer Role Specification Dialog

- For each of the reviewers, select the desired role from its associated Role option menu. When all of the reviewer roles have been specified, click on the "OK" button.

- You may delete reviewers from the Review Package by selecting the reviewers you wish to delete and then clicking the "Delete" button under the Reviewer listbox.

  NOTE: In order for a Review Package to be started, at least one reviewer must be specified.

- If desired, you may add additional Design Objects to the Review Package by clicking on the "Add Design Object" button. The standard file browser will appear and you will be able to add Design Objects as before when the Review Package was created. Note that with the current version of the Review Utility, the review leader may not delete any Design Objects from the Review Package.

- Select the "OK" button to submit the information to the server.

  Once the server accepts the information, each Reviewer in the Reviewer List is notified (via email) of the Review Package.

Starting a Review Package During Creation

As a convenience to the Review Package creator, it is possible to start the Review Package concurrently with the creation of the Review Package. The following steps detail this process.

- Bring up the **Create Review Package** dialog by selecting "Review -> Create" from the menu bar. The dialog shown in Figure 9, "Create Review Package and Start," on page 131 will appear.

  NOTE: This figure shows all the fields completed.

- Fill in the values for the fields as detailed in "Creating a Review Package Without Starting".

- In order to start the Review Package at its creation, the list of reviewers will have to be specified. Reviewer specification may be done through reviewer lists, individual reviewer specification or a combination of both. Please refer to "Starting the Review After Creation" for a detailed description on how to specify reviewers.

- Ensure that the "Close Date" field contains a valid value.

- When all of the information has been entered, click on the "OK" button to submit the information to the server. This will create and start the Review Package. Note that the user which creates the Review Package does not have to be the review leader in order to create and start the package.

  Once the server accepts the information, each reviewer in the Reviewer List is notified (via email) of the Review Package.

Create Review Package	

Project: RDE ▼

Subsystem: Review Utility ▼

Title: Redliner Interface   Review Leader: hickman

Close Date:
(Month/Day/Year) 09/25/95

Design Objects	Reviewer Name	Reviewer Role
ColorSelectorDlg.cc	Rob Hickman	Hardware Engineer
redline.cc	Chris Chrulski	Technical Expert
RedlineTool.cc	Senthil Natarajan	Hardware Engineer
shapeCommentPixmap.cc		
DesignObject.C		
do_shell.C		
do_shell.H		

Add	Delete	Add	Delete	List

Submitter Comments

Please review these design objects

OK	Cancel	Help

FIGURE 9.   Create Review Package and Start

Reviewing a Package

The Review Utility system will notify all reviewers that there is a Review Package requiring their attention. In order for a reviewer to review a package, the following steps must be taken.

- Select the "Reviewer" filter option from the "Package List Type" option menu. A list of all packages for which the user has been specified as a reviewer will appear.

- Select the appropriate Review Package from the list of packages as shown in Figure 10, "Opening Review Package," on page 133.

- Select "Package -> Open" from the menu bar.

FIGURE 10.   Opening Review Package

- This will bring up the **Design Objects** dialog (see Figure 11, "Design Objects Dialog," on page 134).
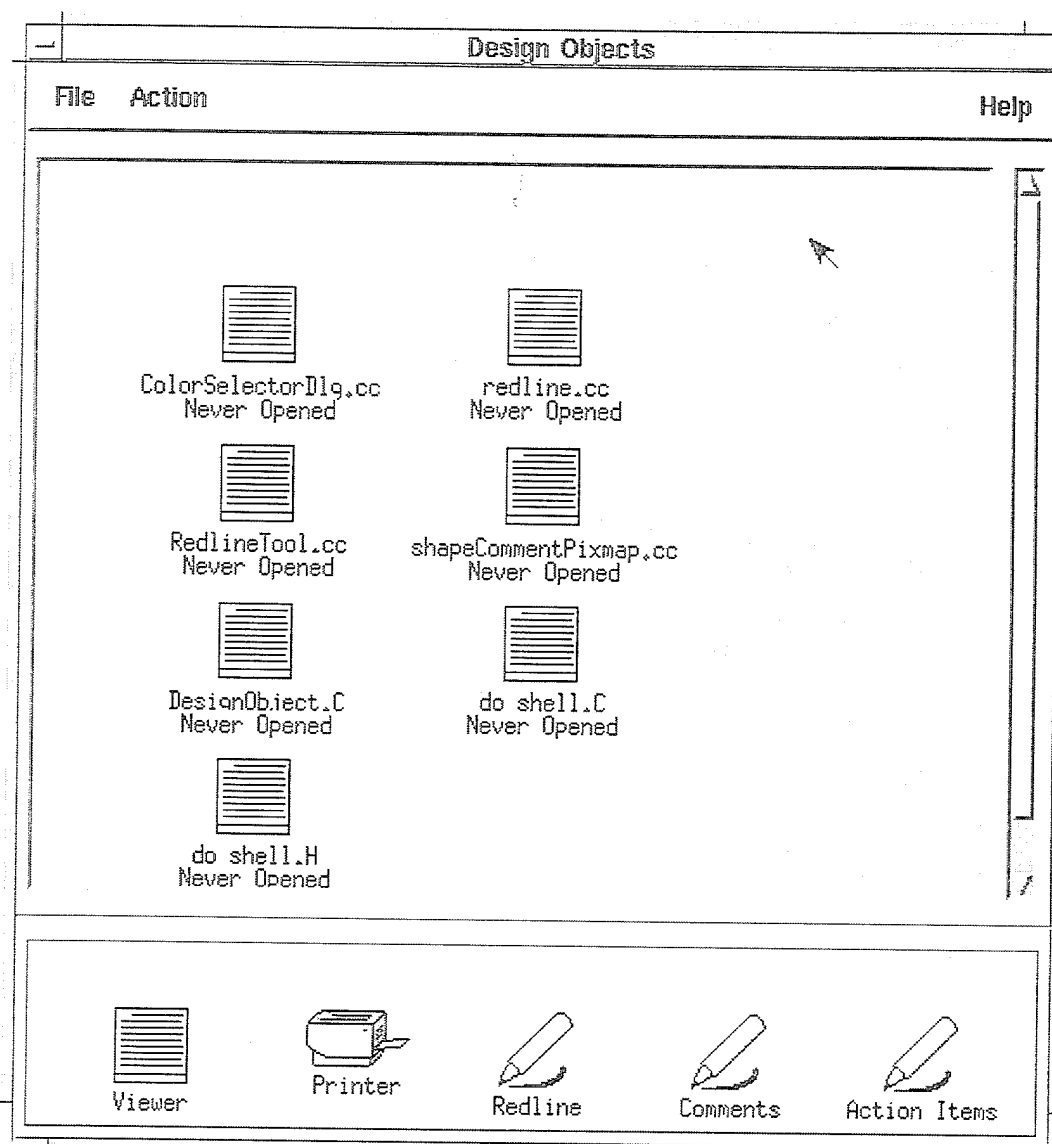
FIGURE 11.   Design Objects Dialog

The menu bar for the **Design Objects** dialog has the following entries:

- ○   **File** - View, Print, Close

- ○   **Action** - Comments..., Action Items..., Redline...

- ○   **Help** - Context Sensitive Help, Help...

To activate any of the Design Object features (Comments, Action Items, View, Print or Redline), click on the icon which is associated with the desired Design Object. When the Design Object has been selected, its title will be displayed in reverse video. After selecting the Design Object, select the desired feature from the menu bar.

In addition to using the menu bar to access these features, the user may also employ "drag and drop" techniques. To use the "drag and drop" technique, click and hold mouse button 2 on the desired Design Object icon. While holding down mouse button 2, drag the Design Object on top of the desired drop target icon. The drop target icons are located at the bottom portion of the Design Objects dialog. There are drop targets for Viewer, Printer, Redline, Comments and Action Items. When the Design Object is over one of the drop targets, an outline around the drop target will appear to provide visual feedback that the drop target has been activated. When the outline appears around the desired drop target, release mouse button 2 to activate the feature for the Design Object.

Figure 12, "Drag and Drop Technique for Launching Design Object Features," on page 136 shows a Design Object that has been dropped over the Redline drop target in order to launch the Redline Tool.
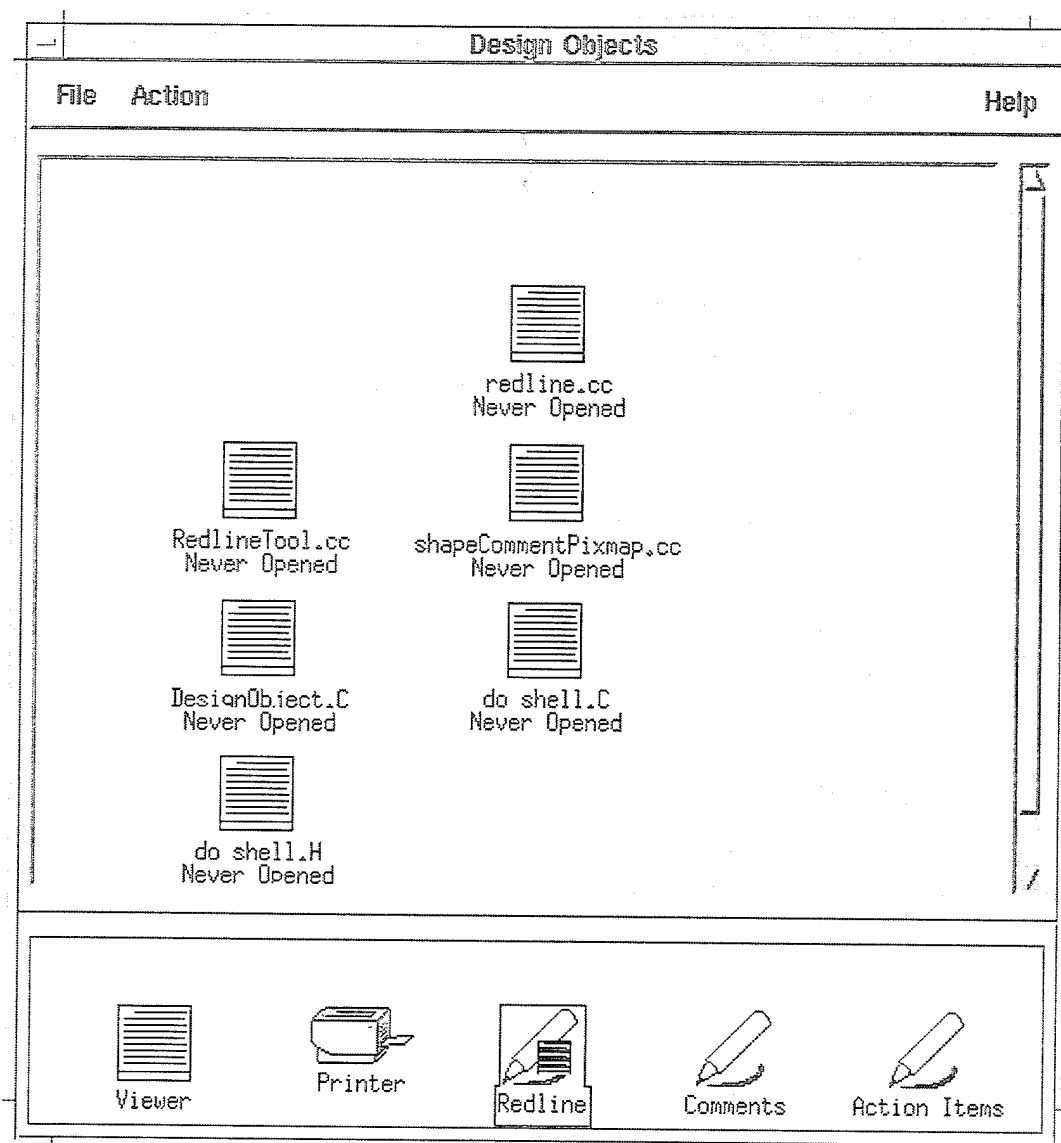
**FIGURE 12.   Drag and Drop Technique for Launching Design Object
Features**

Comments

As part of the review process, the reviewers will be creating comments, editing comments and viewing the comments made by other reviewers. All comment actions are accessed via the Comments dialog. You may bring up the Comments dialog for the desired Design Object using either the Design Objects dialog menu bar or the drag and drop method. The Comments dialog is shown in Figure 13, "Comments Dialog," on page 137.
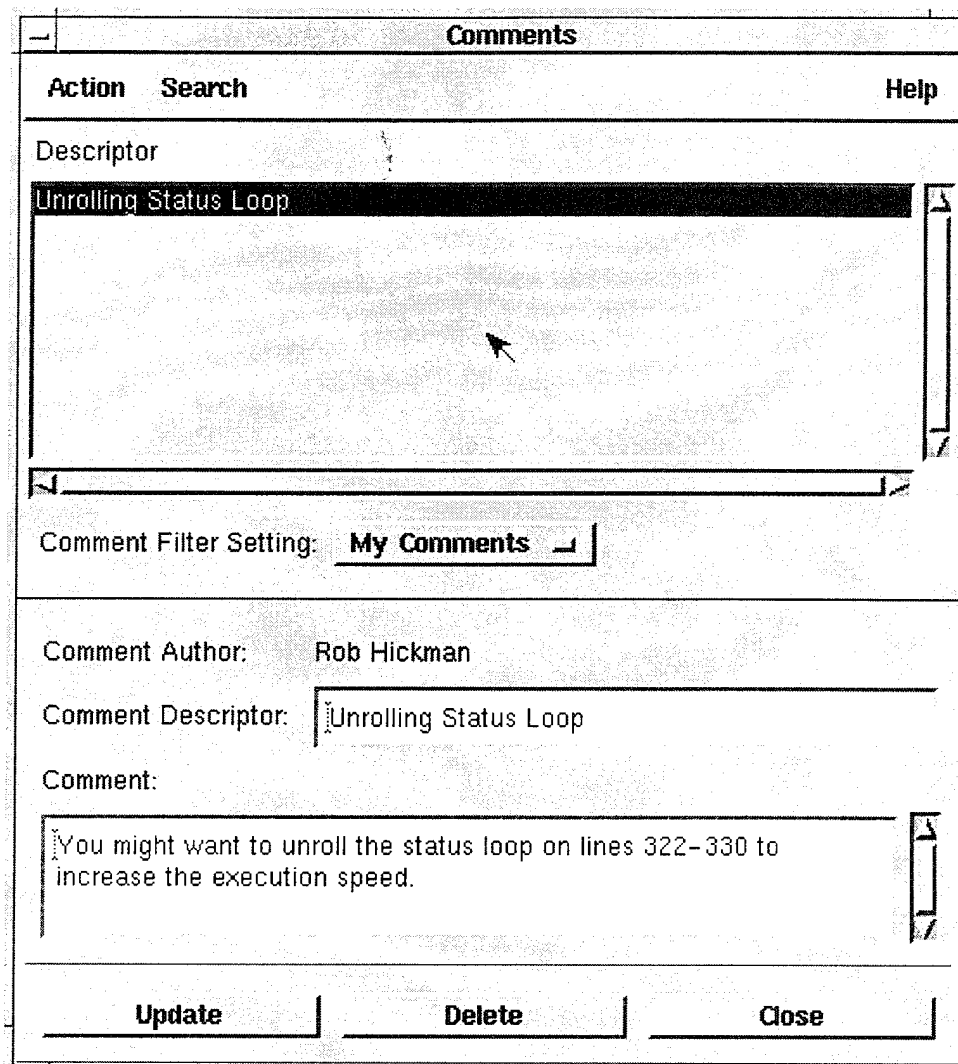
FIGURE 13.   Comments Dialog

The Comments dialog is divided into two halves. The upper half of the dialog contains the menu bar, the Comments listbox and the Comment Filter Setting option bar (this setting can be My Comments, All Comments or Search Results). The lower half of the dialog contains the data fields of the currently selected comment (if applicable) and the comment modification buttons.

The Comments listbox displays all of the comments for the selected Design Object as specified by the Comment Filter Setting value. If

the Comment Filter Setting is set to My Comments, then only your comments are displayed in the listbox. If the Comment Filter Setting is set to All Comments, then every comment that has been entered for the Design Object will be displayed. If the Comment Filter Setting is set to Search Results, then all comments which match the most recent search criteria will be shown.

The menu bar for the Comments dialog has the following entries:

○   **Action** - New..., Update, Delete, Close

○   **Search** - Search...

○   **Help** - Context Sensitive Help, Help...

Creating a New Comment

To create a new comment, perform the following actions.

○   Select "Action -> New..." from the menu bar. This will bring up the Create Comment Form dialog (refer to Figure 14, "Create Comment Form Dialog," on page 139).

FIGURE 14.   Create Comment Form Dialog

- Fill in the "Comment Descriptor" and "Comment" fields. The comment descriptor is the shortname version for the comment by which it will be referenced. The "Comment" field is used to fully describe the comment. Both of these fields are required.

- When the fields have been filled in appropriately, click on the "Submit" button to create the comment.

Note that in order for a reviewer to be able to create a new comment, the following conditions must be met: 1) the user must be listed as a reviewer for the Review Package, 2) the reviewer must not have previously closed the Review Package, and 3) the Review Package must have a package status of opened (i.e. it must not be unopened, closed or cancelled). If all of these conditions are not met, then the "New..." menu bar entry will be grayed out.

Viewing a Comment

To view an existing comment, perform the following actions.

○ Select the desired comment from the Comments listbox. The comment descriptor will be highlighted and its associated fields will be displayed in the lower half of the Comments dialog (see Figure 13, "Comments Dialog," on page 137).

Editing a Comment

To edit an existing comment, perform the following actions.

○ Select the desired comment from the Comments listbox.

○ Update the comment attributes in the lower half of the Comments dialog as desired.

○ Click on the "Update" button to update the comment.

Deleting a Comment

To delete an existing comment, perform the following actions.

○ Select the desired comment from the Comments listbox.

○ Click on the "Delete" button to delete the comment. The user will be prompted for verification on the delete operation.

Note that in order for a reviewer to be able to edit or delete an existing comment, the following conditions must be met: 1) the user must be listed as a reviewer for the Review Package, 2) the reviewer must be the owner of the comment, 3) the reviewer must not have previously closed the Review Package, and 4) the Review Package must have a package status of opened (i.e. it must not be unopened, closed or cancelled). If all of these conditions are not met, then the "Update" and "Delete" buttons will be grayed out.

Searching Comments

To search the comment database, perform the following actions:

○ Select "Search -> Search..." from the menu bar. This will bring up the Enter Search Criteria dialog (refer Figure 15, "Enter Search Criteria Dialog," on page 141).
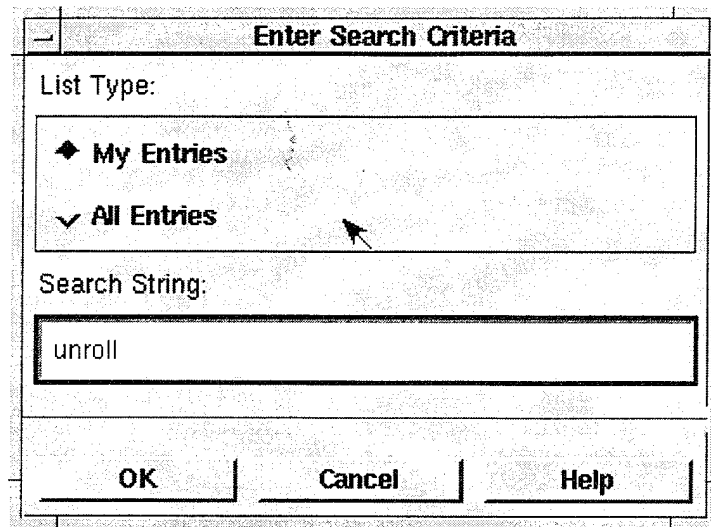
FIGURE 15.   Enter Search Criteria Dialog

- Select the desired "List Type" toggle. If "List Type" is set to My Entries, then the search will only be applied to the comments generated by the current user. If "List Type" is set to All Entries, then the search will be applied to every comment in the database.

- Enter the desired value for Search String. The search string is a regular expression. The search string will be applied to the descriptor and description of each comment that is searched.

- After List Type and Search String have been specified, click on the "OK" button to perform the search. The Comment Filter Setting on the Comments dialog will be switched to Search Results and any comments which matched the search criteria will be shown in the comments listbox (refer to Figure 16, "Comment Search Results," on page 142).

FIGURE 16. Comment Search Results

Action Items

As part of the review process, the reviewers will be creating action items, editing action items and viewing the action items made by other reviewers. All action item functions are accessed via the Action Items dialog. You may bring up the Action Items dialog for the desired Design Object using either the Design Objects dialog menu bar or the drag and drop method. The Action Items dialog is shown in Figure 17, "Action Items Dialog," on page 143.
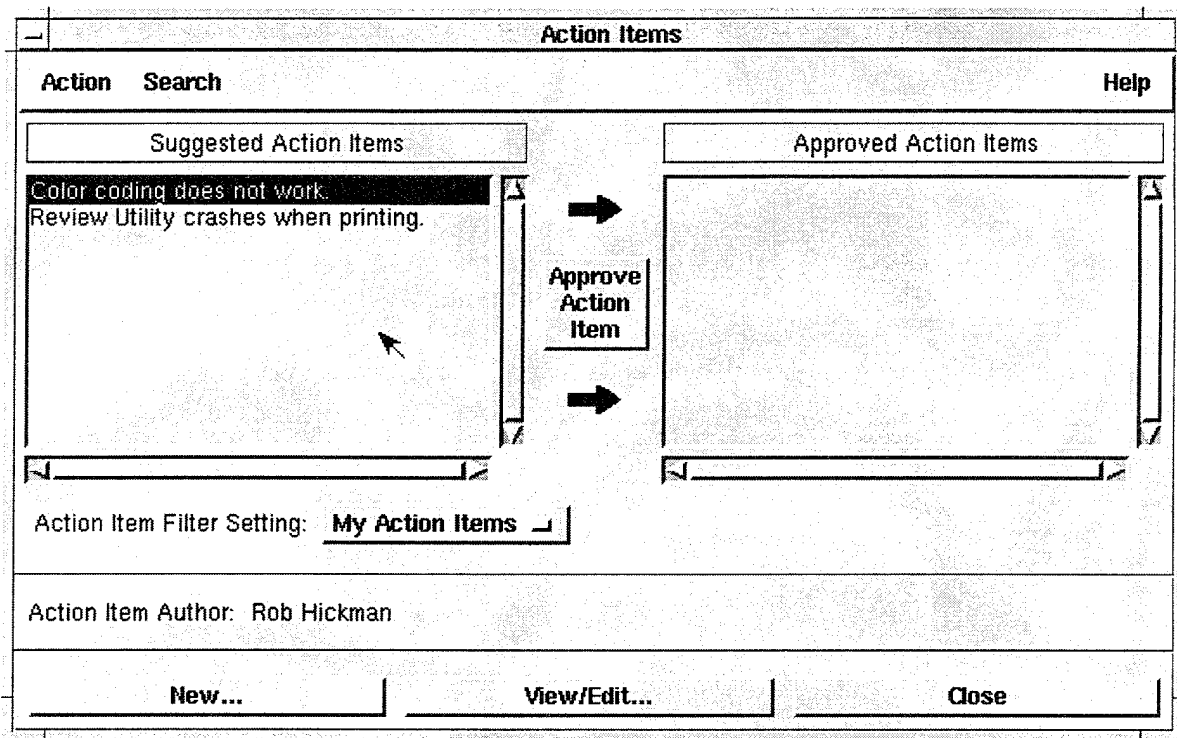
FIGURE 17.   Action Items Dialog

The Action Items dialog contains two listboxes. The listbox on the left contains Suggested Action Items and the listbox on the right contains Approved Action Items.

When reviewers create action items, they are actually suggesting action items. These suggested action items become true action items when the Review Leader approves them. This allows the Review Leader to ensure that multiple action items are not submitted covering the same problem.

As with the Comments dialog, there is an Action Item Filter Setting

option bar which can be set to either My Action Items, All Action Items or Search Results. The bottom part of the dialog has a field to specify the author of the currently selected action item (if applicable) and buttons for creating and editing action items.

Both the Suggested Action Items and the Approved Action Items list-boxes display the action items for the selected Design Object as specified by the Action Item Filter Setting value. If the Action Item Filter Setting is set to My Action Items, then only the reviewer's action items are displayed in the listboxes. If the Action Item Filter Setting is set to All Action Items, then every action item that has been entered for the Design Object will be displayed. If the Action Item Filter Setting is set to Search Results, then all action items which match the most recent search criteria will be shown.

The menu bar for the Action Items dialog has the following entries:

° **Action** - New..., View/Edit..., Close

° **Search** - Search...

° **Help** - Context Sensitive Help, Help...

Creating a Suggested Action Item

To create a suggested action item, perform the following actions.

° An Action Item can be suggested by a reviewer by clicking on the "New..." button at the bottom of the Action Items dialog. This will bring up the New Problem Report dialog shown in Figure 18, "New Problem Report Dialog," on page 146.

° The New Problem Report dialog is very similar to the main dialog for the Problem Report Utility. Please refer to the Problem Report Utility section in this User's Guide for directions and descriptions of the various boxes and buttons.

° When you have entered the appropriate data into the form, click on the "Submit" button to create the suggested action item.

Note that in order for a reviewer to be able to create a suggested action item, the following conditions must be met: 1) the reviewer must not have previously closed the Review Package, and 2) the Review Package must have a package status of opened (i.e. it must not be unopened, closed or cancelled). If all of these conditions are not met, then the "New..." button will be grayed out.

FIGURE 18. New Problem Report Dialog

Viewing an Action Item

To view an action item, perform the following actions.

- Select the action item to be viewed by clicking on its synopsis in the appropriate action item listbox. The selected action item will be highlighted in reverse video.

- Click on the "View/Edit..." button near the bottom of the dialog. This will bring up the **Open Action Item** dialog shown in Figure 19, "Open Action Item Dialog," on page 148.

- When you have finished viewing the action item, click on the "Cancel" button to return to the **Action Items** dialog.

Open Action Item

Class?
- ◆ SW-bug
- ◇ Doc-bug
- ◇ Change Request
- ◇ Support

Priority?
- ◇ Low
- ◇ Medium
- ◆ High

Severity?
- ◇ Non-critical
- ◆ Serious
- ◇ Critical

Category | RDE

Synopsis: Review Utility crashes when printing.

Enter the precise description of the problem:

When selecting "File->Print" from the Design Objects dialog in the Review Utility, a segmentation fault occurs.

Enter the code/input activities to reproduce the problem:

Select "File->Print" from the Design Objects dialog menubar.

Enter the fix or workaround to solve the problem:

Unknown.

Phase Inserted: Coding and Unit Testing

Phase Detected: Coding and Unit Testing

Update   Delete   Cancel   Help

FIGURE 19.   Open Action Item Dialog

Editing a Suggested Action Item

To edit a suggested action item, perform the following actions.

- Select the action item to be edited by clicking on its synopsis in the appropriate action item listbox. The selected action item will be highlighted in reverse video.

- Click on the "View/Edit..." button near the bottom of the dialog. This will bring up the **Open Action Item** dialog shown in Figure 19, "Open Action Item Dialog," on page 148.

- Edit the data fields appropriately. When you have finished editing the action item, click on the "Update" button to submit the new information to the server.

Deleting a Suggested Action Item

To delete a suggested action item, perform the following actions.

- Select the action item to be deleted by clicking on its synopsis in the appropriate action item listbox. The selected action item will be highlighted in reverse video.

- Click on the "View/Edit..." button near the bottom of the dialog. This will bring up the **Open Action Item** dialog shown in Figure 19, "Open Action Item Dialog," on page 148.

- Click on the "Delete" button to delete the action item. The Review Utility will request confirmation on the delete operation.

Note that in order for a reviewer to be able to edit or delete an action item, the following conditions must be met: 1) the selected action item must be a suggested action item (approved action items may not be modified), 2) the reviewer must be the owner of the action item, 3) the reviewer must not have previously closed the Review Package, and 4) the Review Package must have a package status of opened (i.e. it must not be unopened, closed or cancelled). If all of these conditions are not met, then the "Update" and "Delete" buttons will be grayed out.

Approving a Suggested Action Item

The Review Leader can promote an action item from suggested to approved by highlighting the action item and clicking on the "Approve Action Item" button. The approved action item is put into the GNATS database (if it exists). Future tracking of the action item should be performed through the GNATS database.

Searching Action Items

To search the action item database, perform the following actions:

- Select "Search -> Search..." from the menu bar. This will bring up the Enter Search Criteria dialog (refer Figure 15, "Enter Search Criteria Dialog," on page 141). Note that the Search Criteria dialog for performing a search of action items is identical to the Search Criteria dialog for performing a search of comments.

- Select the desired "List Type" toggle. If "List Type" is set to My Entries, then the search will only be applied to action items generated by the current user. If "List Type" is set to All Entries, then the search will be applied to every action item in the database.

- Enter the desired value for Search String. The search string is a regular expression. The search string will be applied to the synopsis and description of each action item that is searched.

- After List Type and Search String have been specified, click on the "OK" button to perform the search. The Action Item Filter Setting on the Action Items dialog will be switched to Search Results and any action items which matched the search criteria will be shown in the action item listboxes (refer to Figure 20, "Action Item Search Results," on page 151).
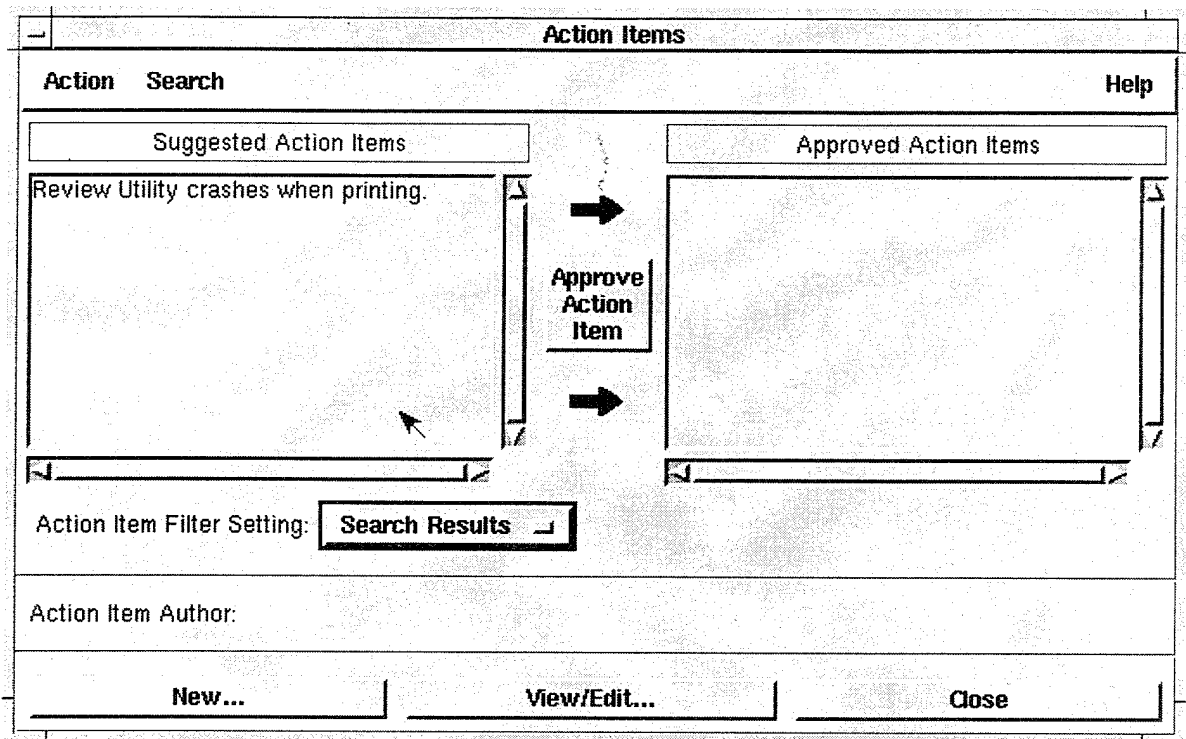
FIGURE 20.   Action Item Search Results

Redlining a Design Object

As part of the review process, the reviewers will be redlining design objects. This includes making new redline marks and reviewing redline marks made by the other reviewers. All redline actions are accessed via the **Redline Tool**. You may bring up the **Redline Tool** for the desired Design Object using either the **Design Objects** dialog menu bar or the drag and drop method. The **Redline Tool** is shown in Figure 21, "Redlining a Design Object," on page 152.

*NOTE: The **Redline Tool** can handle text, GIF and XWD files. Other types of files may not be redlined.*

```
┌─┬─────────────────────────────────────────────────────────────────────┬───┬─┐
│ ⌐│                         ActionItem.h                                │ ◢ │ │
├──┴─────────────────────────────────────────────────────────────────────┴───┴─┤
│  Markup   Colors                                                               │
│ ┌────────────────────────────────────────────────────────────────────────────┐│
│ │ ┌─────┐ ┌─────┐ ┌─────┐ ┌─────┐ ┌─────┐ ┌─────┐                             ││
│ │ │  ╱  │ │  ○  │ │  □  │ │  ]  │ │  A  │ │NOTE │  ▨ hickman ▨ chrisc □ snowman ▨ stupka
│ │ └─────┘ └─────┘ └─────┘ └─────┘ └─────┘ └─────┘                             ││
```

```
0    /*
1    * Module Name  : ActionItem.h
2    * Revision     : [1]  [this will be determined by the TDM]
3    * Creation Date: 12/7/1994
4    * Mod. Date    : [Month/Day/Year]
5    *
6    * Synopsis:
7    * This module defines the ActionItem class.
8    *
9    *
10   * Requirements:
11   *
12   * Change History:
13   * [Rev Date Name]
14   * [What Was Done]
15   *
16   * 1 12/7/1994 chrisc
17   * created the module
18   *
19   * Programmer: Chris Chrulski (chrisc@rassp.rassp.mot.com)
20   * Phone      : (602) 441-3699
21   * Address    : Motorola Government and Systems Technology Group
22   *             M/D H1175
23   *             8201 E. McDowell Road
24   *             Scottsdale AZ 85252-1417
25   *
26   * Copyright (c) 1994 Motorola, Inc.
27   */
28
29                                       Need Comments Here  I
30   #ifndef _ACTIONITEM_H
31   #define _ACTIONITEM_H
32
33   #include <rw/cstring.h>
34   #include <rw/collstr.h>
35   #include <rw/dlistcol.h>
36
37   #include "Reviewer.h"
38
39   typedef enum { UNAPPROVED = 0, APPROVED } AIStatusType;
40
41   typedef struct ActItem_data {
```
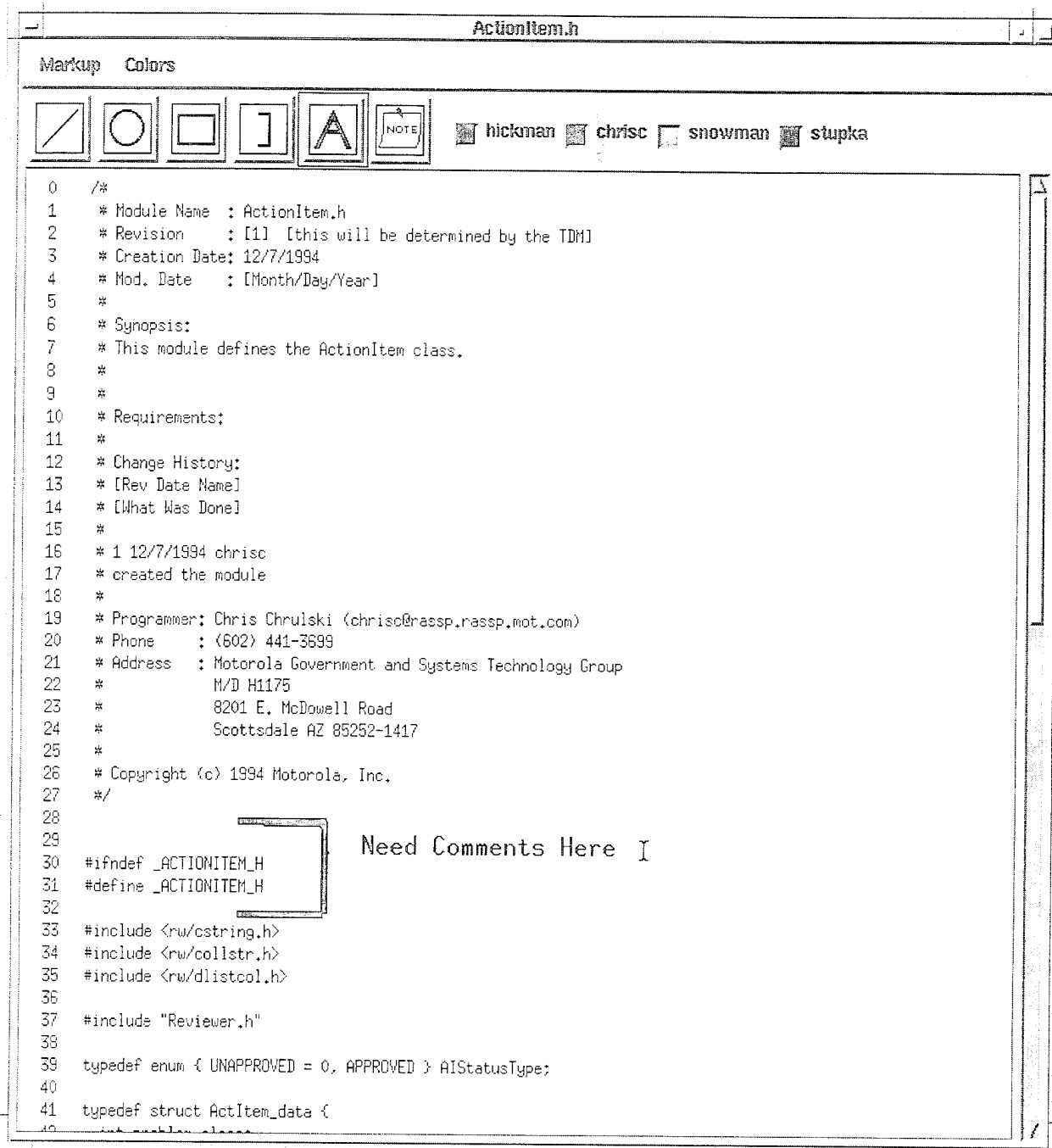
FIGURE 21.   Redlining a Design Object

The **Redline Tool** is divided into two portions. The upper portion contains the menu bar, tool palette and reviewer listing. The lower portion contains the scrolled window which displays the design object which is being redlined.

The menu bar for the **Redline Tool** has the following entries:

- **Markup** - Save, Dismiss

- **Colors** - Assign...

Figure 21, "Redlining a Design Object," on page 152 shows the Redline Tool with the Design Object and a few marks placed on it. The Redline Tool maintains a different color for each reviewer.

If you wish, you may assign colors to the reviewers which are different than the default colors given by the **Redline Tool**. This is done though the use of the **Color Palette** dialog (see Figure 22, "Color Palette Dialog," on page 153). To bring up this dialog, select "Colors -> Assign..." from the menu bar.
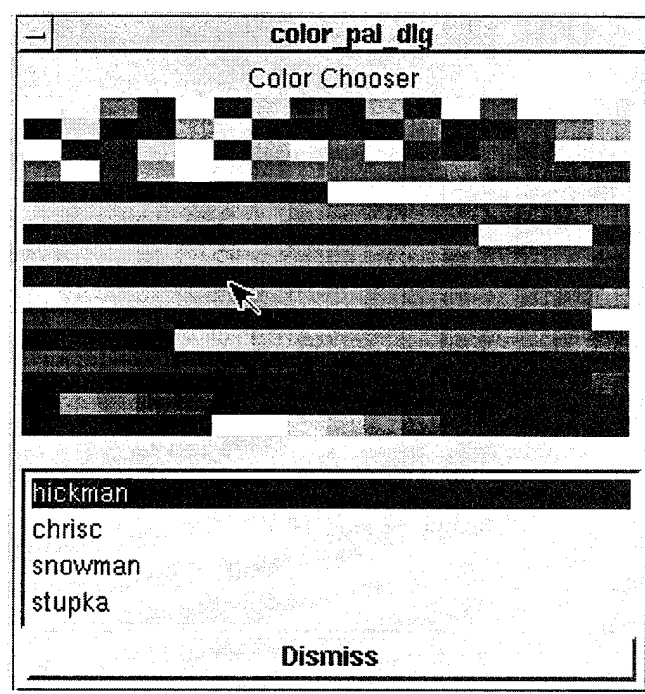

FIGURE 22.   Color Palette Dialog

To assign a color to a reviewer, select the reviewer by clicking on the reviewer's name and then select the desired color in the Color Chooser array. Continue this process for each reviewer you wish to assign a different color. When you have finished, click on the "Dismiss" button.

An individual reviewer's marks can be turned off by clicking on the colored square next to the reviewer's name.

To place a mark on the Design Object, select the desired mark type from the tool palette and place the mark on the Design Object by left-clicking at the location where the mark should be placed. A shift-middle click operation on an existing mark will delete it.

When you have finished marking the object, select "Markup -> Dismiss" from the menu bar. This will save the marks and return to the **Design Objects** dialog.

Additional Actions for Design Objects

In order to help the reviewer with the review task, there are two additional Design Object features available: viewing and printing.

Viewing a Design Object

You may view a Design Object by using either the **Design Objects** dialog menu bar or the drag and drop method. This will launch the appropriate viewer application for the selected Design Object.

The viewer application is determined by the file extension of the Design Object. The file, $RDE_HOME/etc/view_tool_table, specifies a one-to-one mapping between file extension and viewer application. If no mapping exists for the Design Object, then the default viewer application is launched.

The Design Object that is displayed in the viewer application is actually a temporary, read-only version of the Design Object that is deleted when the viewer application is exited.

Printing a Design Object

This feature is not currently implemented.

Closing a Review Package

After finishing the reviewing task, the reviewer must inform the system that he or she is done with the review. This is accomplished by performing the following tasks.

- Highlight the desired Review Package in the main Review Utility dialog (Figure 1, "Review Utility Startup Dialog," on page 119).

- Select "Package -> Close" from the menu bar. The user is then presented with the dialog shown in Figure 23, "Closing a Review Package," on page 155.

Close Review Package

Reviewer Disposition:

◆ **Accepted**

✓ **Conditionally Accepted**

✓ **New Review**

✓ **Redo Module**

Additional Comments:

Good work

FIGURE 23.   Closing a Review Package

- The **Close Review Package** dialog allows the reviewer to set his or her "disposition" (one of "Accepted", "Conditionally Accepted", "New Review", or "Redo Module"). There is also a section for "Additional Comments." The reviewer should place comments which pertain to the disposition and the entire Re-

view Package here.

Stopping the Review

After all reviewers have reviewed the package, or after the time has expired on the review, the Review Leader should stop the review. To stop a review, the following actions are performed:

○ Ensure that the value of the "Package List Type" option menu is "Leader".

○ Designate the Review Package to be stopped by selecting the package title in the listbox of the main Review Utility dialog.

○ Select "Review -> Stop" from the menu bar. The dialog shown in Figure 24, "Stopping the Review," on page 156 will appear.

○ Enter the review summary in the "Review Leader Summary" field. This field is required.

○ Click on the "OK" button to stop the review.

The review has now been stopped and email has been sent to the author to notify him or her that the review has been stopped. This notification also includes the review leader's summary.
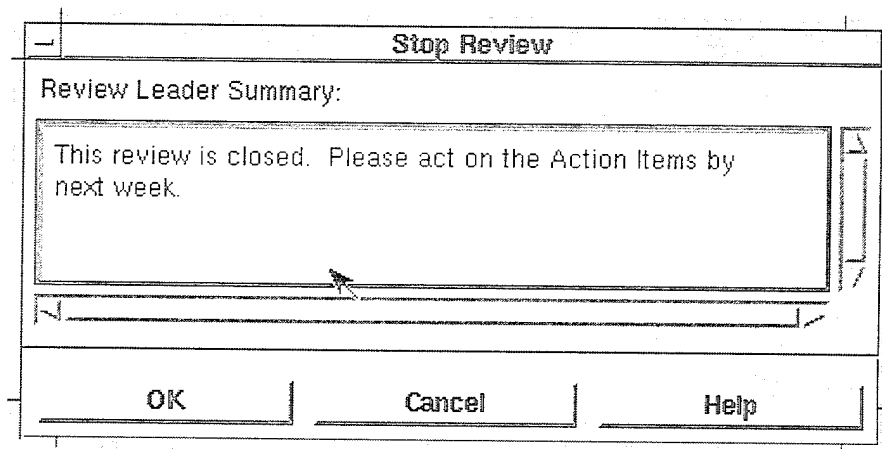

FIGURE 24.   Stopping the Review

Cancelling the Review

If necessary, the Review Leader may cancel a review. To cancel a review, the following actions are performed:

- Select the "Leader" filter option from the "Package List Type" option menu on the main Review Utility dialog (shown in Figure 1, "Review Utility Startup Dialog," on page 119). This will display all of the Review Packages for which the user is the Review Leader.

- Select the Review Package to be cancelled from the list of packages displayed.

- Select "Review -> Cancel" from the menu bar. The dialog shown in Figure 25, "Cancelling the Review," on page 157 will appear.

- Enter the reason for cancelling the review. This field is required.

- Click on the "OK" button to cancel the review.

The review has now been cancelled and email has been sent to the author to notify him or her that the review has been cancelled. This notification also includes the reason why the review was cancelled.
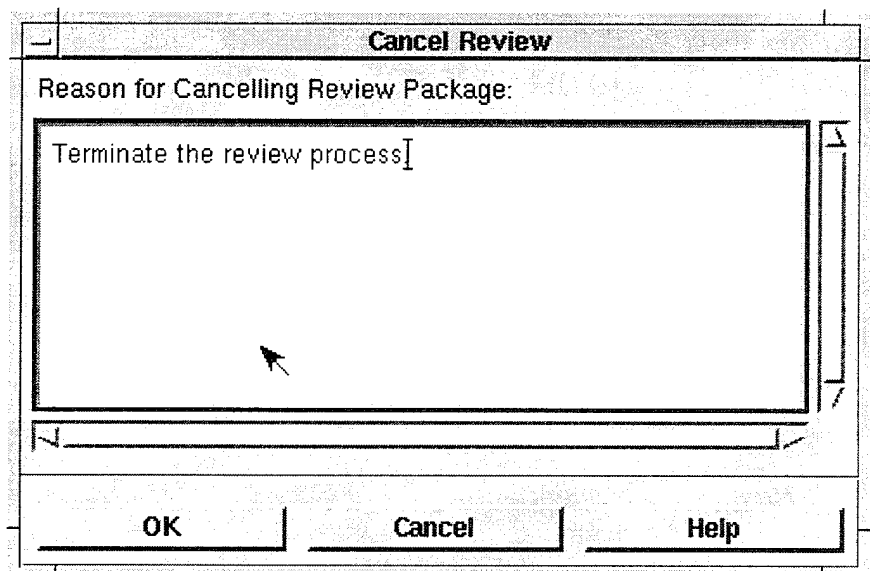

FIGURE 25. Cancelling the Review

Additional Features

The Review Utility has five additional features which are provided to help the user with the technical review process: 1) Reviewer Listing, 2) Review Summary, 3) Named Package Filtering, 4) Default Reviewer Lists, and 5) Message Center. All of these features are supplemental and are not necessary to perform a review.
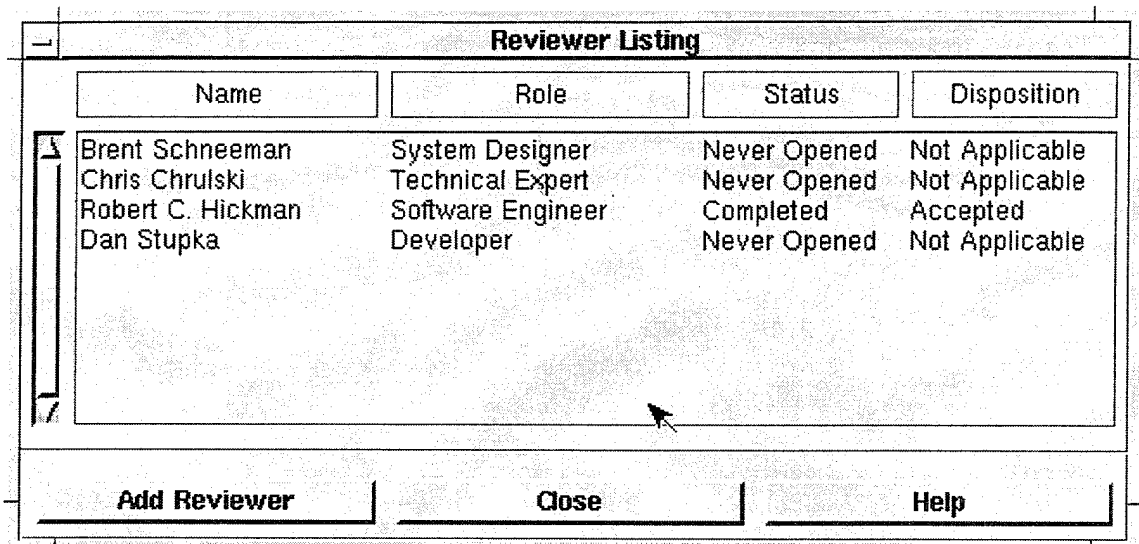
Reviewer Listing

The **Reviewer Listing** dialog provides several areas of functionality.

- It gives a listing of all of the reviewers that have been assigned to the currently selected Review Package (if any).

- It displays important information associated with each of the reviewers. This includes reviewer role, reviewer status and reviewer disposition.

- It provides the means to add additional reviewers to a Review Package that has already been started.

To display the **Reviewer Listing** dialog, perform the following actions.

- Designate the Review Package for which you wish to see the Reviewer Listing by selecting the package title in the listbox of the main Review Utility dialog.

- Select "Review -> Reviewer List" from the menu bar. The dialog shown in Figure 26, "Reviewer Listing Dialog," on page 159 will appear.

FIGURE 26.   Reviewer Listing Dialog

To add reviewers to a review using the **Reviewer Listing** dialog, click on the "Add Reviewer" button. The **Add Reviewers** dialog (shown in Figure 6, "Add Reviewers Dialog," on page 127) will appear and the user will be able to add reviewers as previously described.

Note that in order for a user to be able to add a reviewer to a Review Package, the following conditions must be met: 1) the user must be a reviewer for the Review Package, the Review Leader or the Review Submitter, and 2) the Review Package must be opened (i.e. it must not be unopened, closed or cancelled).

Review Summary

The **Review Summary** dialog provides the user a convenient means of viewing reviewer and leader summaries for a Review Package. This dialog is most useful for Review Packages which have been stopped or cancelled by the Review Leader. There are several items of information that this dialog provides to the user.

• It displays the Review Leader for the currently selected Review Package and gives the Review Leader summary (if applicable).

• It gives a listing of all of the reviewers associated with the currently selected Review Package.

• It displays important information associated with each of the reviewers. This includes reviewer role, reviewer status and re-

viewer disposition.

○ It displays the Reviewer Summary for the currently selected re-
viewer. The Reviewer Summary will only be applicable for re-
viewers whose disposition is not "Not Applicable".

*NOTE: The Review Summary dialog provides a superset of the
information given by the Reviewer Listing dialog. However,
you may not add reviewers to a review package through the Re-
view Summary dialog.*

To display the **Review Summary** dialog, perform the following ac-
tions.

○ Designate the Review Package for which you wish to see the
Review Summary by selecting the package title in the listbox of
the main Review Utility dialog.

○ Select "Review -> Review Summary" from the menu bar. The
dialog shown in Figure 27, "Review Summary Dialog," on
page 161 will appear.

```
┌──────────────────────────────────────────────────────────────────┐
│ ─                        Review Summary                            │
├──────────────────────────────────────────────────────────────────┤
│ Leader:  Rob Hickman                                               │
│                                                                    │
│ Leader Summary:                                                    │
│ ┌────────────────────────────────────────────────────────────┐ ▲ │
│ │ This review is finished.  Please work on the Action Items.  │   │
│ │ They should be resolved by 9/30/95.  Thanks.                │   │
│ │                                                             │ ▼ │
│ └────────────────────────────────────────────────────────────┘   │
│                                                                    │
│   ┌─────────────┬──────────────┬─────────────┬─────────────┐      │
│   │    Name     │     Role     │   Status    │ Disposition │      │
│ ▲ ├─────────────┼──────────────┼─────────────┼─────────────┤      │
│   │ Rob Hickman │ Hardware Eng │ Completed   │ Accepted    │      │
│   │ Chris Chrulski │ Technical Expert │ Forced Closed │ Not Applicable │
│   │ Senthil Natarajan │ Hardware Engineer │ Forced Closed │ Not Applicable │
│ ▼ │             │              │             │             │      │
│   └─────────────┴──────────────┴─────────────┴─────────────┘      │
│                                                                    │
│ Reviewer Summary:                                                  │
│ ┌────────────────────────────────────────────────────────────┐ ▲ │
│ │ Good work.                                                  │   │
│ │                                                             │   │
│ │                                                             │ ▼ │
│ └────────────────────────────────────────────────────────────┘   │
│ ┌──────────────┐  ┌──────────────┐  ┌──────────────┐              │
│ │    Shrink    │  │    Close     │  │    Help      │              │
│ └──────────────┘  └──────────────┘  └──────────────┘              │
└──────────────────────────────────────────────────────────────────┘
```

FIGURE 27.   Review Summary Dialog

The Review Summary dialog is composed of three sections. The top section displays the Review Leader and the Leader's summary for the Review Package. The middle section lists all of the reviewers that have been assigned to the Review Package. The bottom section displays the Reviewer Summary for the currently selected reviewer.

To view a reviewer's summary, select the desired reviewer in the listbox. When you have finished viewing the Review Summary, click

on the "Close" button to dismiss the dialog.

> *NOTE: Disregard the "Shrink" button on the Review Summary dialog. This feature is not implemented in this build of the Review Utility.*

Named Package Filtering

In order to help the user organize the listing of review packages, the Review Utility provides the means to create custom filters. Currently, these named package filters provide the ability to discriminate review packages based on three attributes: 1) project name, 2) subsystem name and 3) review package status.

Note that there are four filters already built into the Review Utility accessed via the "Package List Type" option menu. Depending on its setting, the review package listing with be filtered as follows:

- "Package List Type"=Reviewer - Filters the review package listing so that only review packages for which the user has been assigned as a reviewer are displayed.

- "Package List Type"= Submitter - Filters the review package listing so that only review packages which the user created are displayed.

- "Package List Type"= Leader - Filters the review package listing so that only review packages for which the user is specified as the review leader are displayed.

- "Package List Type"=All - No filtering is performed on the review package listing. All review packages are displayed.

Named package filters allow for additional filtering in conjunction with the "Package List Type" filtering.

All named package filter actions are accessed via the **Named Package Filters** dialog. You may bring up the **Named Package Filters** dialog by selecting "Filter -> Show Named Filters" from the menu bar of the main Review Utility dialog (see Figure 1, "Review Utility Startup Dialog," on page 119). The **Named Package Filters** dialog is shown in Figure 28, "Named Package Filters Dialog," on page 163.
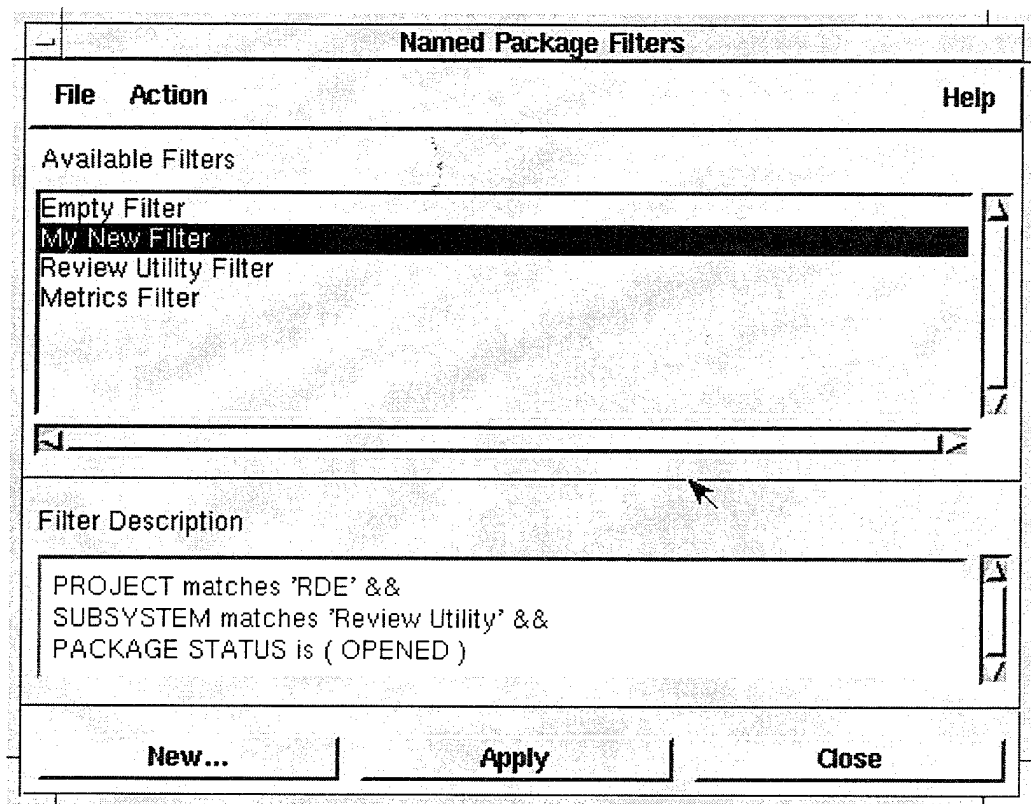
FIGURE 28.   Named Package Filters Dialog

The **Named Package Filters** dialog is divided into two halves. The upper half of the dialog contains the menu bar and the Available Filters listbox. The lower half of the dialog contains the filter description of the currently selected filter (if applicable) and the package filter action buttons.

The Available Filters listbox displays all of the named package filters that exist for the user. Named package filters are kept in the user's home directory in the file, ".ReviewTool-filter-history". Any time a change is made to the named package filters list, the updated list is saved to this file. Therefore, the user never has to explicitly save his or her named package filters.
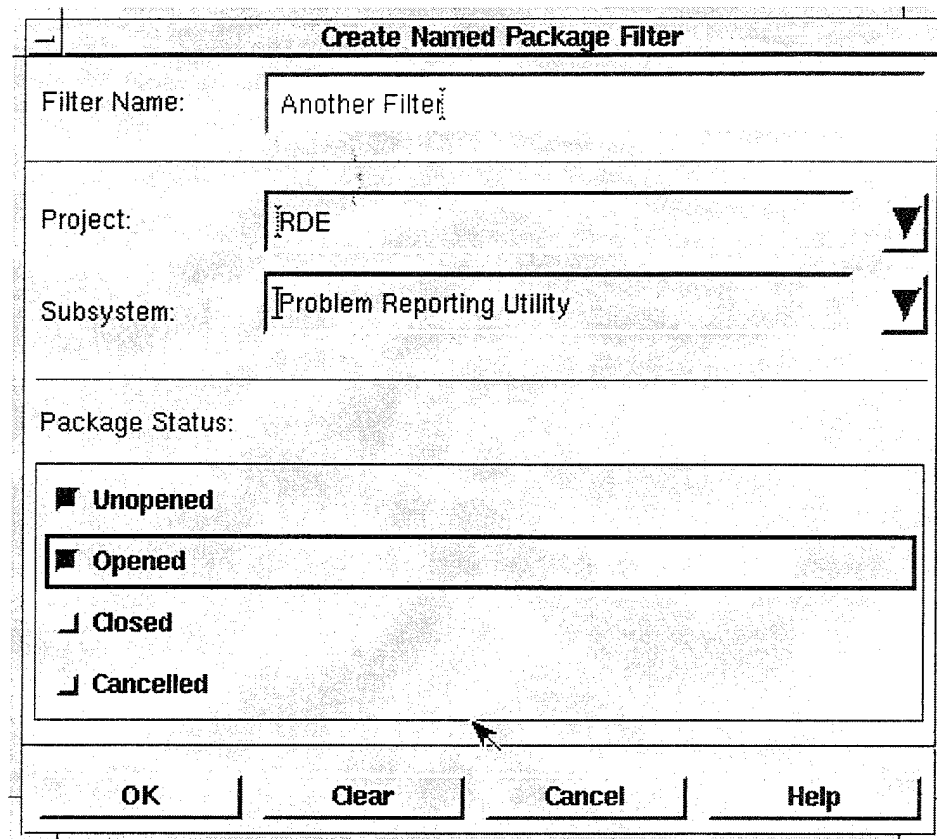
The menu bar for the **Named Package Filters** dialog has the following entries:

- File - New..., View/Edit..., Delete, Close

- Action - Apply

- Help - Context Sensitive Help, Help...

Creating a Named Package Filter

To create a new named package filter, perform the following actions.

- Select "File -> New..." from the menu bar (alternatively you may click on the "New..." button near the bottom of the dialog). This will bring up the Create Named Package Filter dialog (refer to Figure 29, "Create Named Package Filter Dialog," on page 165).

FIGURE 29.   Create Named Package Filter Dialog

○ Fill in the "Filter Name" field. This will be the name by which the filter will be referred. This field is required.

○ Fill in the "Project" and "Subsystem" combo boxes appropriately. Note that these combo boxes are editable. You may select a value from the drop-down list or you may directly enter a value into the text field. The Review Utility supports named package filters which contain regular expressions. Therefore, the "Project" and "Subsystem" may contain regular expressions to perform more complex filtering. These values are combined as a boolean AND expression in the package filtering.

○ Click on the appropriate toggles for "Package Status". These values are combined as a boolean OR expression in the package filtering.

○ If you wish to clear all of the data fields, click on the "Clear" button.

○ When the fields have been filled in appropriately, click on the "OK" button to create the named package filter.

Editing a Named Package Filter

To edit a named package filter, perform the following actions.

○ Designate the filter you wish to edit by clicking on its name in the Available Filters listbox.

○ Select "File -> View/Edit..." from the menu bar. This will bring up the Edit Named Package Filter dialog (refer to Figure 30, "Edit Named Package Filter Dialog," on page 167).

FIGURE 30.   Edit Named Package Filter Dialog

- All of the data fields will be filled in with the current filter's attributes. Edit these fields as desired. If necessary, refer to the section on creating a new named package filter for descriptions of the data fields.

- If you wish to clear all of the data fields, click on the "Clear" button.

- If you decide that you do not want to edit the filter, click on the "Cancel" button and the filter will remain unchanged.

- When you have finished editing the named package filter, click on the "OK" button to post the changes.

Deleting a Named Package Filter

To delete a named package filter, perform the following actions.

- ○ Designate the filter you wish to delete by clicking on its name in the Available Filters listbox.

- ○ Select "File -> Delete" from the menu bar. The Review Utility will ask for confirmation on the delete.

Applying a Named Package Filter

To perform any filtering on the review package listing, a named package filter must be applied. To apply a named package filter, perform the following actions.

- ○ Designate the filter you wish to apply by clicking on its name in the Available Filters listbox.

- ○ Select "Action -> Apply" from the menu bar (alternatively you may click on the "Apply" button near the bottom of the dialog).

This will set the selected named package filter as the current filter. The name of the current package filter is displayed on the main Review Utility dialog (see Figure 1, "Review Utility Startup Dialog," on page 119) in the middle third of the screen immediately below the "Package List Type" option bar.

Filtering will only take place if named package filtering is on. If named package filtering is off, no filtering will take place and the name of the current package filter will be grayed out.

Toggling Named Package Filtering On and Off

Named package filtering may be turned on and off as desired. To toggle named package filtering, perform the following actions.

- Select "Filter -> Named Filtering" from the menu bar of the main Review Utility dialog (see Figure 31, "Turning Named Package Filtering On," on page 170).

- If named package filtering was previously off, it will now be turned on. This is designated by a red toggle appearing to the left of "Filter -> Named Filtering". If a named package filter was previously specified, then the review package listing will be regenerated and the named package filter name will no longer be grayed out.

- If named package filtering was previously on, it will now be turned off. This is designated by the absence of a red toggle to the left of "Filter -> Named Filtering". The review package listing will be regenerated. If a named package filter was previously specified, the named package filter name will be grayed out.

FIGURE 31. Turning Named Package Filtering On

Default Reviewer Lists

To make the process of creating and starting a review easier for the user, the Review Utility provides the means to create and maintain default reviewer lists. Default reviewer lists are reviewer lists associated with a key project/subsystem identifier.

Default reviewer lists are very useful when there is a core set of reviewers for a particular subsystem on a project. Rather than specifying this core set of reviewers each time a review is started, the user can do it once as a default reviewer list. Thereafter, whenever a re-

view package is created or started with the same project/subsystem combination, the reviewers that comprise the default reviewer list will automatically be assigned to the review package.

All default reviewer list actions are accessed via the **Default Reviewer Lists** dialog. You may bring up the **Default Reviewer Lists** dialog by selecting "Reviewer List -> Show Reviewer Lists" from the menu bar of the main Review Utility dialog (see Figure 1, "Review Utility Startup Dialog," on page 119). The **Default Reviewer Lists** dialog is shown in Figure 32, "Default Reviewer Lists Dialog," on page 171.
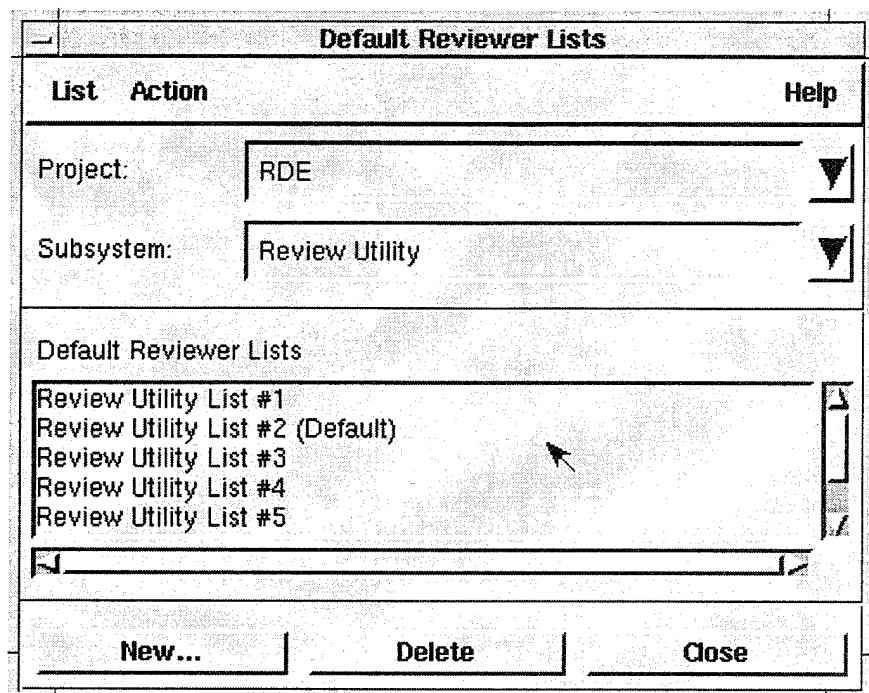

FIGURE 32.   Default Reviewer Lists Dialog

The **Default Reviewer Lists** dialog is divided into two halves. The upper half of the dialog contains the menu bar and the Project and Subsystem combo boxes. The lower half of the dialog contains Default Reviewer Lists listbox and the default reviewer list action buttons.

The Default Reviewer Lists listbox displays all of the default reviewer lists for the current project/subsystem setting. The "default" default reviewer list for the project/subsystem can be determined by

observing which default reviewer list has the designation, **(Default)**, to the right of its name.

Default reviewer lists are specified by a user and exist only for that user. A user's default reviewer lists are not stored in the Review Utility's database and therefore are not accessible to others.

Default reviewer lists are kept in the user's home directory in the file, ".ReviewTool-default-rev-list". Any time a change is made to the collection of default reviewer lists, the updated collection is saved to this file. Therefore, the user never has to explicitly save his or her default reviewer lists.

The menu bar for the **Default Reviewer Lists** dialog has the following entries:

- **List** - New..., Edit..., Delete, Close

- **Action** - Make Default

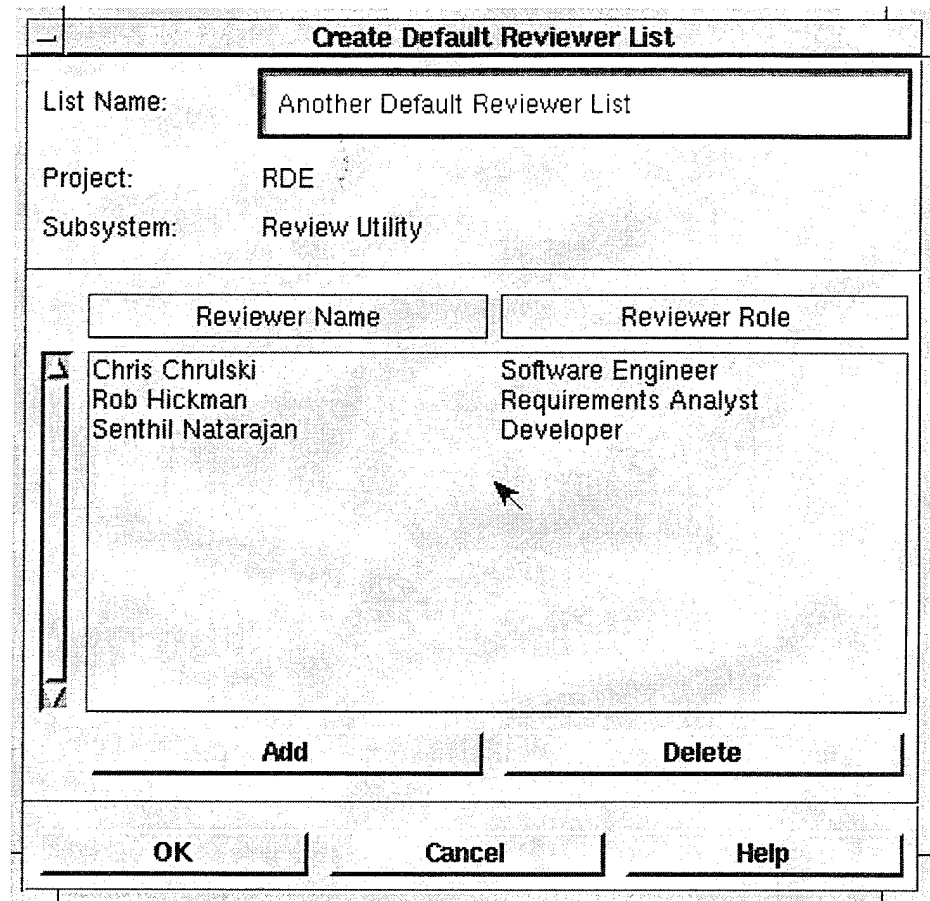- **Help** - Context Sensitive Help, Help...

Creating a Default Reviewer List

To create a new default reviewer list, perform the following actions.

- Select a value for the Project combo box. This will specify the project for which the default reviewer list will be applicable.

- Select a value for the Subsystem combo box. This will specify the subsystem for which the default reviewer list will be applicable.

  NOTE: You will be unable to create a new default reviewer list unless valid values have been specified for both Project and Subsystem.

- Select "List -> New..." from the menu bar (alternatively you may click on the "New..." button near the bottom of the dialog). This will bring up the Create Default Reviewer List dialog (refer to Figure 33, "Create Default Reviewer List Dialog," on page 173).

FIGURE 33. Create Default Reviewer List Dialog

° Fill in the "List Name" field. This will be the name by which the default reviewer list will be referred. This field is required.

° To add reviewers to the default reviewer list, click on the "Add" button. The Add Reviewers dialog (shown in Figure 6, "Add Reviewers Dialog," on page 127) will appear. You may use this dialog to add reviewers as previously described (refer to the section "Creating/Starting a Review Package" for detailed instructions for reviewer specification).

° If you wish to delete reviewers from the default reviewer list, select the desired reviewers in the reviewer listbox and then click on the "Delete" button. The Review Utility will ask for confirmation on the delete operation.

° When the List Name field has been filled in appropriately and the reviewers for the default reviewer list have been specified, click on the "OK" button to create the default reviewer list.

Editing a Default Reviewer List

To edit a default reviewer list, perform the following actions.

° Designate the default reviewer list you wish to edit by clicking on its name in the Default Reviewer Lists listbox.

NOTE: You must specify values for both Project and Subsystem before any default reviewer lists appear in the Default Reviewer Lists listbox.

° Select "List -> Edit..." from the menu bar. This will bring up the Edit Default Reviewer List dialog (refer to Figure 34, "Edit Default Reviewer List Dialog," on page 175).

FIGURE 34. Edit Default Reviewer List Dialog

- All of the data items will reflect the current default reviewer list's attributes. Edit these items as desired. If necessary, refer to the section on creating a new default reviewer list for descriptions of the data items.

- If you decide that you do not want to edit the default reviewer list, click on the "Cancel" button and the default reviewer list will remain unchanged.

- When you have finished editing the default reviewer list, click on the "OK" button to post the changes.

Deleting a Default Reviewer List

To delete a default reviewer list, perform the following actions.

- ○ Designate the default reviewer list you wish to delete by clicking on its name in the Default Reviewer Lists listbox.

  NOTE: You must specify values for both Project and Subsystem before any default reviewer lists appear in the Default Reviewer Lists listbox.

- ○ Select "List -> Delete" from the menu bar (alternatively you may click on the "Delete" button near the bottom of the dialog). The Review Utility will ask for confirmation on the delete.

Making a Default Reviewer List the "Default"

For each group of default reviewer lists on a per project/subsystem basis, there must be one list which is designated as the "default" default reviewer list. The default list will be the default reviewer list that is automatically added to a review package when the package is created or started.

To make a default reviewer list the default, perform the following actions.

- ○ Designate the default reviewer list you wish to make the default by clicking on its name in the Default Reviewer Lists listbox.

  NOTE: You must specify values for both Project and Subsystem before any default reviewer lists appear in the Default Reviewer Lists listbox.

- ○ Select "Action -> Make Default" from the menu bar. You will receive visual confirmation of this action by verifying that the desired default reviewer list has the designation, **(Default)**, after its name in the Default Reviewer Lists listbox.

Message Center

The Message Center dialog allows the user to quickly check several items of information pertaining to the current context of the Review Utility. It also is used to display low-priority status messages (all high-priority status messages are issued through full application modal message boxes).

The Message Center dialog provides the following information:

- The Project name for the currently selected Review Package (if applicable).

- The Subsystem name for the currently selected Review Package (if applicable).

- The Package Title of the currently selected Review Package (if applicable).

- The Design Object name for the currently selected Design Object (if applicable).

- The last low-priority message issued by the Review Utility.

To display the Message Center dialog, perform the following actions.

- Select "Options -> Show Message Center" from the menu bar of the main Review Utility dialog (see Figure 1, "Review Utility Startup Dialog," on page 119). The dialog shown in Figure 35, "Message Center Dialog," on page 177 will appear.

- To hide the Message Center dialog, select "Options -> Hide Message Center" from the menu bar of the main Review Utility dialog.
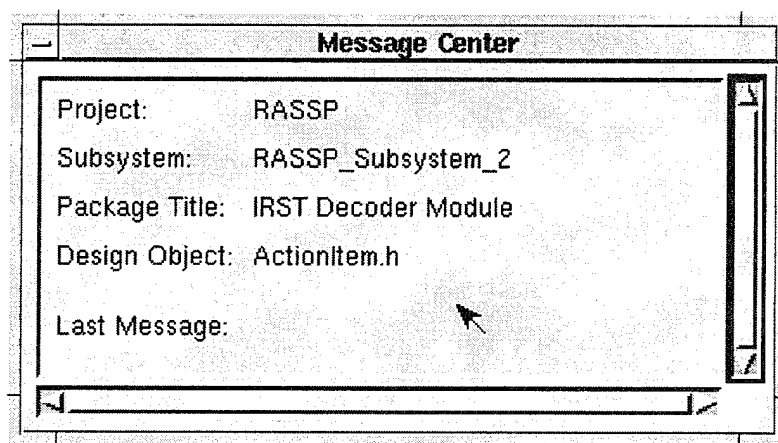

FIGURE 35.   Message Center Dialog

Chapter 10: The Task Manager

1.0 Background

Although there is an abundance of COTS tool that serve the area of Work flow management, there currently exists no COTS tool that deals with the area of task identification and execution. This work represents filling this operational need in a manner that will allow collaboration over a distributed area with task specific items while maintaining a level of situational awareness never before present.

This work is the direct result of efforts put forth by the RASSP Design Environment Prototype (RDEP) team and our thanks go out to them.

This document also reflects the current Alpha release of the product and will be updated with subsequent releases.

2.0 Task Manager Description

The Task Manager allows users from program managers down to engineers direct access and execution of the tasks they have to work on. It allows for the arbitrary dissemination of tasks into subtasks and allows assignment of each of these subtasks to valid users of the system.

The Task Manager supports situational awareness by providing a graphical representation of all the tasks that a user/manager is responsible for. The possibility of "one slipping through the cracks" is reduced to near zero.

The Task Manager supports task execution by providing direct access and tool launch of all data objects associated with a given task. Also, the Task Manager will allow for default or non-default tool launching based on user input. Furthermore, the Task Manager will maintain automatic metrics collection for a variety of metrics that will be described later in this document.
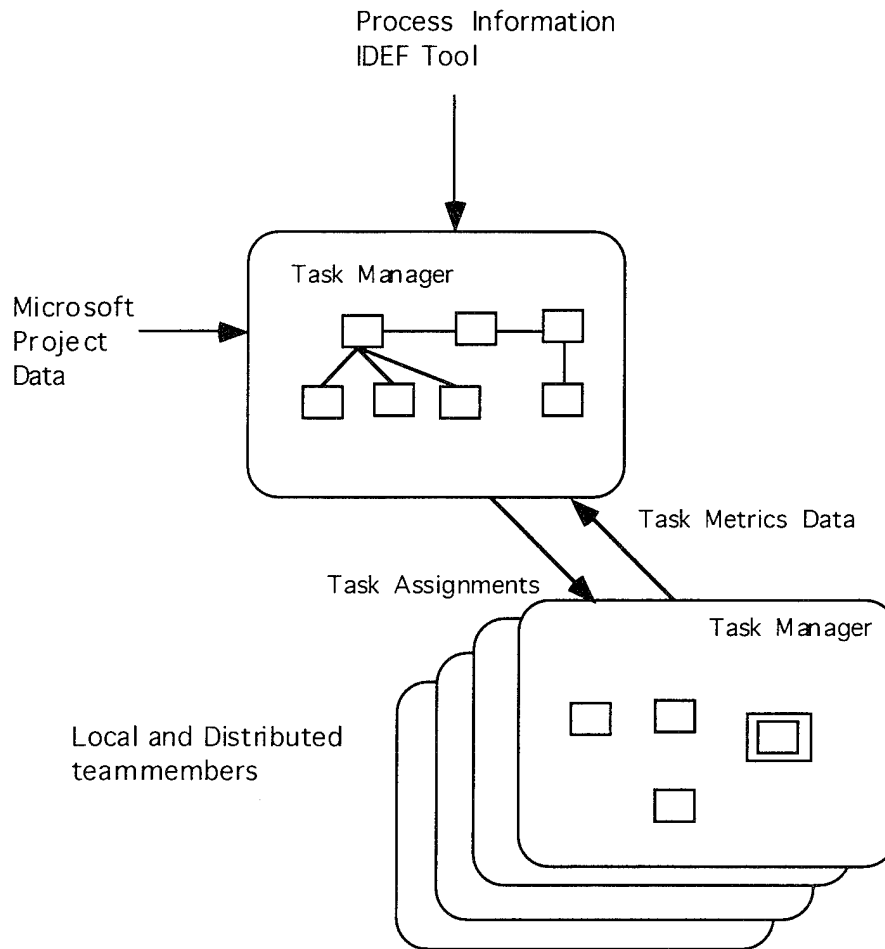
3.0   Task Manager Functionality

Figure 1 shows the top level functionality of the Task Manager. It will take input from either a Microsoft project file or from an IDEF process tool file and generate a set of tasks representative of the data contained therein. (Please see the section on importing for more information).

The user may then add information pertaining to the tasks and assign them to individuals for execution. Once a task has been assigned to a user, that user will be notified of this new task and that user's database will be updated to reflect the new task.

The Task Manager allows each user to customize their tool in order to support them in the work they do without having to change the process of their work significantly. The TM will allow a user to defined a set of default tools relating to each task type and object type so that when a user wishes access to a data object related to a task, a simple mouse click will launch the tool in the user supplied default tool. The TM will also allow the user to launch the data object in any other tool which has been identified as a tool capable of launch a specific data object type.

FIGURE 1.   Task Manager Functionality

Process Information
IDEF Tool

Microsoft
Project
Data

Task Manager

Task Assignments

Task Metrics Data

Local and Distributed
teammembers

Task Manager

4.0 Task Manager Basics

5.0 The User Interface

5.1 Pop-Up and Pull-Down Menus

The menus that are located on the screens will either pop-up or pull-down to display command and data within the Task Manager. Point to a menu and click once to display the commands or data within that menu. Most commands have keyboard shortcuts, which are shown cia an underscore on the command or to the right of the command.

There are three ways to select an option in a menu:

- Point to the menu you want to display, click and hold on the menu, drag the mouse over an option to highlight it, and release the mouse button.
- Point to the menu you want to display, click once to display the menu, and click the desired option.
- Click the menu to display it, and type the keyboard shortcut.

5.2 Mouse Techniques

We use the following conventions when referring to mouse clicks:

- Click means click the left button unless otherwise noted.
- Double-click means click the left button twice.

To select an item, position the cursor over it and click.

5.3 Scroll Bars

The Task Manager has both vertical and horizontal scroll bars to display information that does not fit into a single display window. There are three ways to use a scroll bar:

- Click on the arrow at either end of the bar to move the display one row or column at a time.

- Click on the scroll bar, hold down the mouse and drag the bar up, down, left, or right.

- Click the mouse in the empty space above, below, left, or right of the bar to move the display one screenful in the direction of your click.

5.4 Slider Bars

There are several instances of slider bars within the Task Manager used to change the type of information displayed on the screen. This is usually associated with levels of detail that are displayed in reference to hierarchical information in a single window. There are three ways to use a slider bar:

- Click the mouse on either side of the slider bar to cause it to move one level to the right or left.

- Click and hold on the slide bar while moving the mouse to the right or left to cause it to change based on the amount moved.

- Click the middle mouse button on either side of the slider bar to move the slider bar to the selected location.

5.5 Buttons

Buttons appear in the TM as raised areas on the screen. Buttons are activated by clicking on them. Their labels describe the action that will take place when the button is activated.

5.6 Text Fields

Text fields appear in the Task Manager as indented areas on the screen.

6.0 Getting Started

7.0 Execution Begins

This chapter will cover the basics of using the Task Manager for the first time.

The Task Manager will arrive as a gziped tar file. Download this file in the correct format for your system and move the tar file to the location that the tool will reside in.

Once the file has been stored in the correct directory, unzip the file with the command: "gunzip TM_vxxx_yyy.tar.gz" where xxx is the current version number and yyy is either SOL for Solaris version or SUN for SunOs. This will unpack the file and ready it to be untarred. The next command is: "tar -xvf TM_vxxx_yyy.tar". This will create several directories and unload the files into them. The executables will be stored in the ./bin directory.

To launch the tool, either change to the TM binary directory, or execute a tool launch from the desktop to start up the task manager. If launching will occur from the command line, the command is "tm". This will bring up all executables needed for the Task Manager to operate properly.

8.0   Logging in

Each time a new TM session begins the user is required to log in. This feature will be removed in later versions but was necessary for the alpha version of the software.

The first time that the Task Manager is executed, there will be no users in the database. It will be required at this time to create a new user or users. Please note that the only way to create new users in the TM database currently, is through this log-in interface. Therefore, before a user can be selected as the person a task is assigned to, their user name will have to be added in the login screen.

To add a person to the user database, simply enter their user name into the field provided and click on the button labeled "Add User". Each time a new user is added to the database, the pop-up menu at the top of the login screen will be updated with the latest user.

To log into the Task Manager, either type the users name in the field provided or select the user from the user pop-up menu and hit the button labeled "Log In".

9.0   First Time Browser Screen

Once the user has logged in, the login screen will disappear and be replaced with the Task Browser screen. This will initially be empty due to the fact that there are no tasks in the database. As the user adds tasks to the database that are assigned to that user, this list will begin to populate. For more information on the Task Browser screen, refer to chapter 4.

10.0   Task Manager Elements

This chapter will cover all the elements contained within the Task Manager and will provide information as to their functionality.
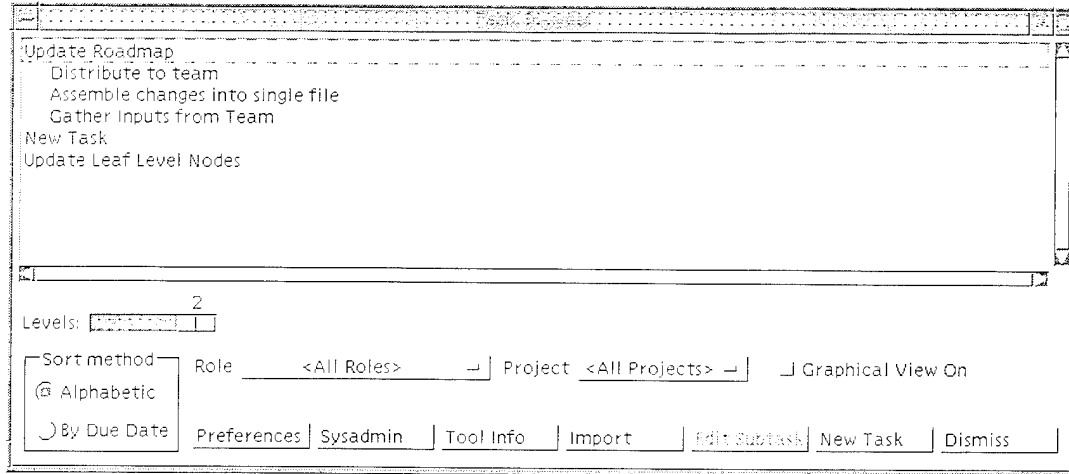
11.0   General Information

The Task Manager is currently made up of two separate views of the data. The main method for viewing and managing tasks is through the "textual" view. This view provides a textual listing of all tasks and provides methods for new task creation and other items. The other view provided in this version of the Task Manager is the "graphical" version. This is an early implementation of where the TM is headed in the future. The graphical view provides an Iconic representation of all tasks that are contained within the system.

Most of the documentation in this chapter will cover the "textual" view of the TM but there will be sections covering the graphical representation later in this chapter.

12.0   Task Browser

This window is contains the main task information presented in the Task Manager. It acts as a starting point for all other TM functions. The Task Browser window can be seen in figure 2.

FIGURE 2.   Task Browser

```
┌─────────────────────────────────────────────── Task Browser ─────────────────────────────────────┐
│ Update Roadmap                                                                                     │
│     Distribute to team                                                                             │
│     Assemble changes into single file                                                              │
│     Gather Inputs from Team                                                                         │
│  New Task                                                                                           │
│  Update Leaf Level Nodes                                                                            │
│                                                                                                    │
│                                    2                                                                │
│  Levels: [          ]                                                                               │
│  ┌Sort method┐  Role     <All Roles>    ⌐│  Project  <All Projects> ⌐│   ⌐│ Graphical View On      │
│  │ⓐ Alphabetic│                                                                                     │
│  │◯By Due Date│ Preferences│ Sysadmin  │ Tool Info │ Import    │ Edit Subtask│ New Task  │ Dismiss  │
└────────────────────────────────────────────────────────────────────────────────────────────────┘
```

At the top of the window is the task presentation area. This is where the tasks that a user is responsible for will appear. In the above example, there are several tasks that contain subtasks. Subtasks are represented by indentations on the text screen and are parented to the task above the subtask that is not indented. The task presentation area is a scrolling window that will allow for arbitrary levels of abstraction and length.

Below the task presentation area is a slider bar that manipulates the level of views into the task hierarchy that is currently presented in the task presentation level. By moving the slider bar to the left or right, the user will change the view into the task viewing area based on levels of abstraction. A level one view will only present root level tasks; those tasks with no direct parent. A level two view will show all root level tasks and subtasks of root level tasks. This dissemination will occur to an arbitrary depth.

In the lower left-hand corner of the Task Browser is an area that will sort the view of the tasks in the task presentation area. This section is currently non-functional and the sorting methods presented in the push-buttons are a sample of the different types of sorting methods that are being investigated. There may be additional methods of viewing the data in future versions that are not presented in this area currently.

Next to the sorting area of the Task Browser is a pop-up menu called "Roles". This menu is a carry over from the RDEP and is currently non-functional. It is still not clear whether this feature will be carried over into the operational Task Manager. What this menu would provide is a means of viewing only those tasks for a single user that are of a specific role.

Next to the role menu is the Project menu. This pop-up menu is also non-functional but will allow a user to view only the tasks associated with a given project.

The next item on that same line is a check box labelled "Graphical View On". This button enables the graphical representation that will be described later in this document.

Along the bottom of the screen lies the buttons that allow the user to gain access to the other sections of the Task Manager. Those sections will be discussed in order from left to right in the following sections.

13.0 Preferences

The Task Manager Preferences screen is used to set the user defined data items for the tasks and data items a user is dealing with. This screen will allow the user to set default settings for newly created tasks and to define promotion hierarchies for task types. Some of these functions may eventually be restricted to system administration functions, but for now they are publicly available. Most all attributes of tasks are available for tailoring by the user.

The Task Manager Preferences window is partitioned into groups of similar functions as seen in figure 2. Each of these functions is described below in detail.

FIGURE 3.   Task Manager Preferences Window
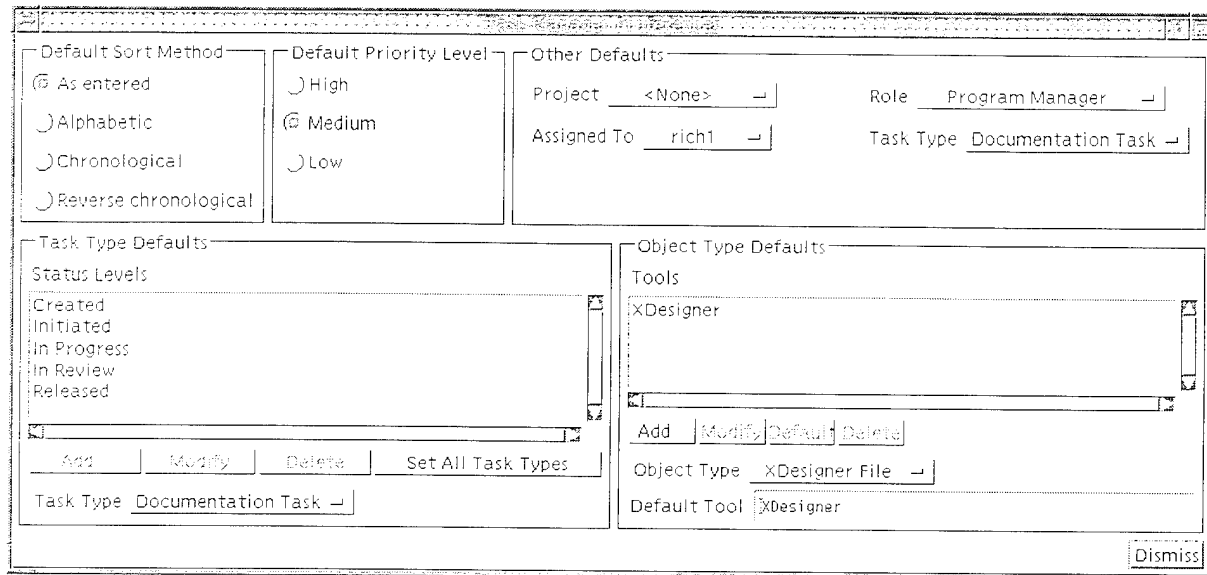
In the upper left hand corner of this window resides the default task sorting method check-boxes. This feature is currently non-functional but will operate as soon as the sorting functions of the Task Browser are completed. These check boxes will allow the user to define the default sorting method that will be used in the Task Browser. This feature may go away depending on user feedback related to the Task Browser showing the sorting of tasks in the last selected mode, rather than a default mode each time.

Next to the Default Sorting Method box lies the Default Priority level box. This group of check-boxes is used when creating new tasks. The priority set at this preference level will be the one that is automatically selected at the time of task creation. This priority level can then be changed if the user so chooses.

In the upper right-hand corner of the Preferences window resides the Other Defaults box. This box contains multiple pop-up menus that provide default sections for the Project a task gets assigned to, the Role that a task will be associated with, the person that is assigned the task, and the type of the task. Once a user has set these preferences, each time a new task is created,

these values will be initially inserted into those slots. The user may then change them as to their liking.

The lower left-hand section of this window contains a box labelled "Task Type Defaults". What this box deals with is the definition of task promotion hierarchy defaults based on task type. This data is initially populated with a promotion scheme but can be specialized based on task type. This function is currently available to all users but will most likely be restricted in the future to on system administration type people.

In the lower right-hand corner resides the box that is used to define default tools for object types. When a new object type is defined, it will be necessary to define a default launching tool. This is also the area that is used to redefine which tool will be used for an object type. In the example in the figure above, the object type "X Designer file" has been associated with the application X-Designer.

There is currently no method to directly relate other tools to data objects for usage beyond the default tool. Once a tool has been defined as a default tool for a data type, it gets stored in the database for use later when changing the tool to launch in. In future versions of the tool, there will be a section included to relate tools to data types in a more user-friendly manner.

14.0 Sysadmin

The system administration page has been implemented in order to provide various object definitions that were performed in other area's of the RDEP but were needed within the stand-alone tool.

The types of database objects that a user may create in this window are: Roles, Objects, Task Types, and Object Types. When one of these items is added or modified, it is saved into the database and used throughout the rest of the system.

This window represents functionality that would most likely not be available to the general public but would reside in a restricted area only for system administrators. Due to the fact that the system is currently limited to single user mode, this function will be available to all users.

FIGURE 4.   TM Type Definitions
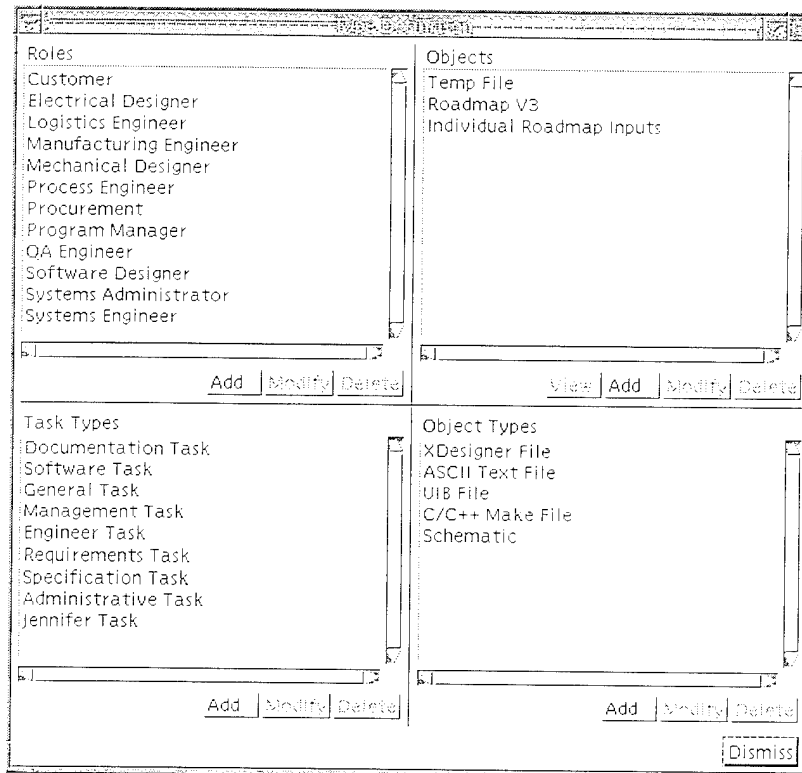
15.0 Tool Info

This window is a representation of the tool metrics that the task manager collects automatically. The data contained in this window includes: tool start, tool stop time, project, task, data item, user, information about launching (default or other), and will include more information

as it is identified. This screen was added late in development in response to user requests for this type of information and will be updated later based on feedback.

FIGURE 5.   Tool Launch Data window
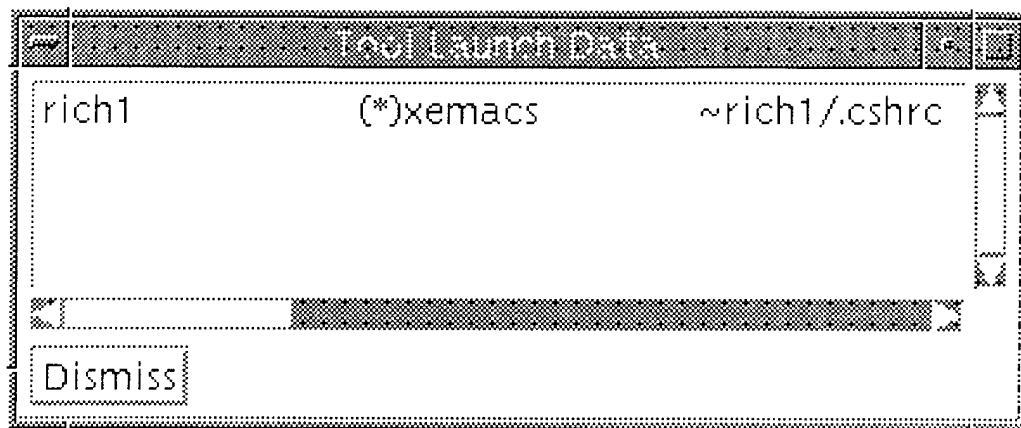
16.0 Import

The current incarnation of the task manager contains the ability to import files from Microsoft Project. The figure below shows the screen that the Task Manager uses to perform this function. In order to import files into the Task Manager, it will be required to take the Microsoft Project data and save it into a ".csv" format. Once the file is in the correct format, use this window to read it into the TM database.

The current version of the Task Manager only creates tasks for activities defined in the Microsoft Project file that have man-power allocated to it. This could be changed in future versions based on user feedback.

When the TM creates the tasks from the imported file, each task is created as a root level task. These tasks can then be broken down into subtasks at the users discretion.

FIGURE 6.   Task Import Window
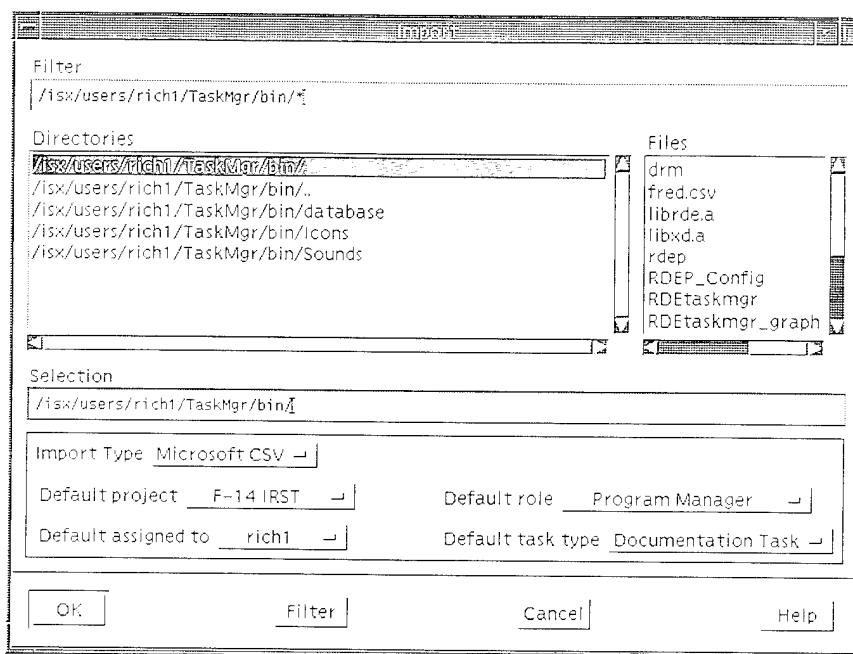
17.0 Edit Subtask

The Task Editor window is the main location within the Task Manager where task data is entered and edited. Each of the different functions of the Task Editor screen will be discussed below.

At the top of this window contains the area that represents links into other sections of the RDE. The only link that is currently functional on this screen is the "Task Parent" link. When

a subtask is loaded into this window, this link will change the context of the editing session to that of the parent. The product structure link and the workflow link will be added in the future.

Each of the items in the middle of the window are used to define elements of the task that will be saved to the database. Wherever a button exists beside the text entry field, that data is stored in the database and is accessible through a pop-up window.

At the bottom left of the screen is the area that represents the current task hierarchy. This window will be updated as the tasks are created and deleted.

At the bottom right of this window is the text entry area that is used for entering the task description. This field is saved in the database and is used to communicate addition information about the task to the user.

At the bottom of the screen are buttons that serve a variety of functions. The only ones that are currently functional are the buttons labelled: more info, Save, Add Subtask, and Dismiss. The help and remove buttons will be functional in later versions.

FIGURE 7.   Task Editor Window

When a user clicks the "More Info" button, an additional screen of information will be shown, as shown in the figure below. This screen contains addition task meta-data that will be stored in the database. This screen is also the location where the inputs and ouputs of a given task are defined.

These input and output areas on the screen will also allow the user to launch a specific data file based on data that has been entered on the file. For example, in the window below there exists an input for the Roadmap V3. By clicking on that input and clicking the "View" button, the TM will launch the tool defined for that input and load the tool with the specified data.

FIGURE 8.   Task Editor Additional Information

18.0 New Task

When a user clicks the "New Task" button from the Task Browser, the TM will create a new database record and bring up the Task Editor with the task name of "New Task". Then the user merely edits the task as described above in the Task Editor section.

19.0 Task Manager Graphical View

This section covers the graphical task representation within the Task Manager. Due to its current maturity level, most of the complex functions of the task manager exist only in the textual

view as described above. However, there exists enough functionality in the graphical view to be useful.

The graphical representation is made up of three separate windows. These are the History window, the Bookmark window, and the Tasks Window. The history window contains an iconic representation of each of the levels of tasks hierarchy that a user is currently in to. This allows a better understanding of the level of the subtask that a user may be interested in.

Another window on the graphical representation is the bookmark window. This window contains icons that will take a user to a specific point within the task hierarchy that the user has identified as important. A task gets placed into this window when a user defines a bookmark within the task window.

The Task Window is the main task execution window within the Task Manager. It is this window that the user will use to launch tasks, define bookmarks, and call up addition task information.

When a task has a subtask defined for it, a special graphical representation of a box surrounding that task is drawn to indicate the existence of the decomposition. By clicking on these decomposed tasks, the user changes the context of the tasks window to that of the children of that task. The change also adds a new icon to the history window for traversal purposes.

To launch a default data object with the default tool of that object type, all the user has to do is single click on the icon. All data objects of a task are accessible through a right mouse click, as-well-as the ability to launch tasks in another non-default tool.